Kubernetes DaemonSet Jobs & Cron Jobs

DaemonSet

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

DaemonSet
Pod

DaemonSet

Some typical uses of a DaemonSet are:

- running a cluster storage daemon on every node
- running a logs collection daemon on every node
- running a node monitoring daemon on every node

DaemonSet
Pod

CLARUSWAY©
WAY TO REINVENT YOURSELF

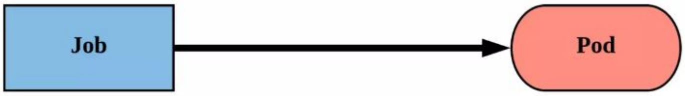
Kubernetes Jobs

A Job creates one or more **Pods** and will continue to retry execution of the Pods until a specified number of them successfully terminate

CLARUSWAY©
WAY TO REINVENT YOURSELF

Kubernetes Jobs

- Job controller ensures one or more pods are executed and successfully terminate.
- Will continue to try and execute the job until it satisfies the completion and/or parallelism condition.
- Pods are **NOT** cleaned up until the job itself is deleted.



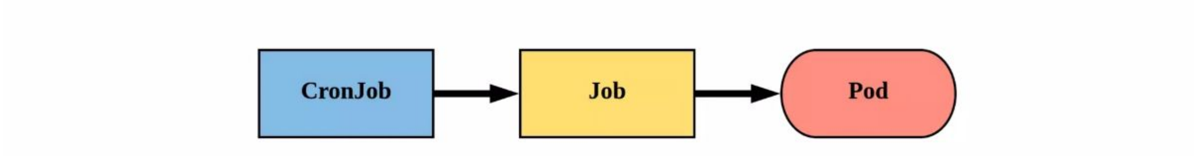
Kubernetes Jobs

- backoffLimit: The number of failures before the job itself is considered **failed**.
- completions: The total number of successful completions desired.
- parallelism: How many instances of the pod can be run concurrently.
- spec.template.spec.restartPolicy: Jobs only support a restartPolicy of type Never or OnFailure.

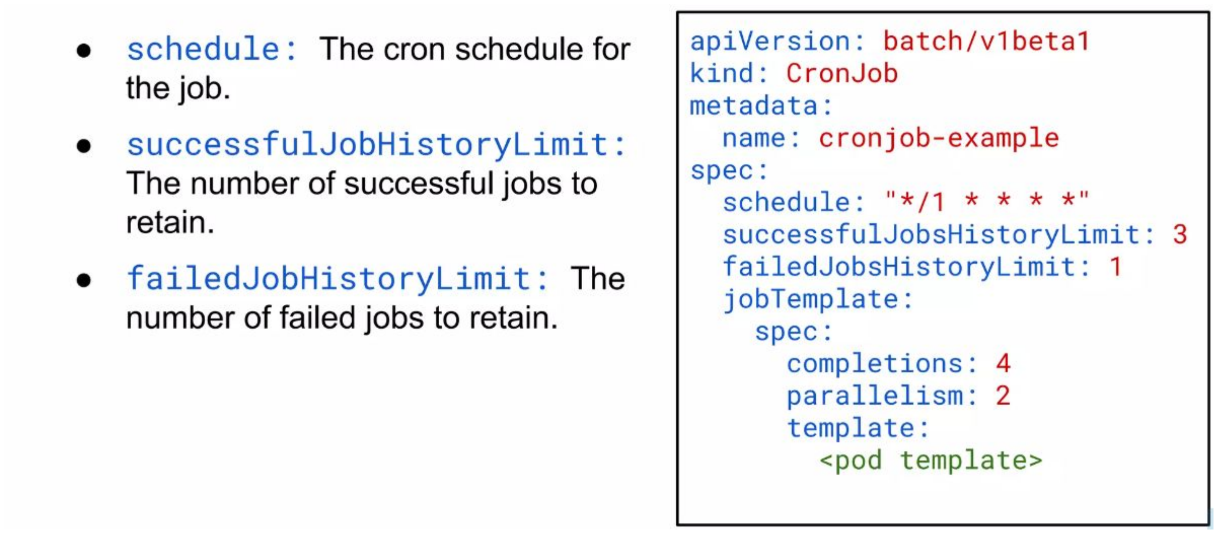
```
apiVersion: batch/v1
kind: Job
metadata:
  name: job-example
spec:
  backoffLimit: 4
  completions: 4
  parallelism: 2
  template:
    spec:
      restartPolicy: Never
    <pod-template>
```

Kubernetes CronJob

CronJob is meant for performing regular scheduled actions such as backups, report generation, and so on. One CronJob object is like one line of a crontab (cron table) file on a Unix system. It runs a Job periodically on a given schedule, written in Cron format.



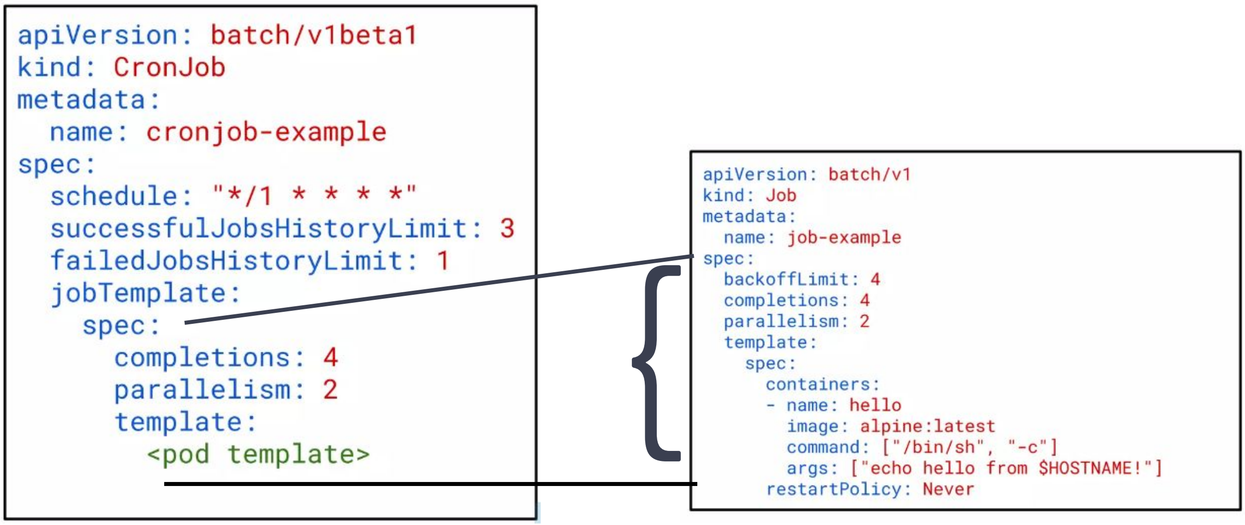
Kubernetes CronJob



- schedule: The cron schedule for the job.
- successfulJobHistoryLimit: The number of successful jobs to retain.
- failedJobHistoryLimit: The number of failed jobs to retain.

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: cronjob-example
spec:
  schedule: "*/1 * * * *"
  successfulJobsHistoryLimit: 3
  failedJobsHistoryLimit: 1
  jobTemplate:
    spec:
      completions: 4
      parallelism: 2
      template:
        <pod template>
```

Kubernetes CronJob



```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: cronjob-example
spec:
  schedule: "*/1 * * * *"
  successfulJobsHistoryLimit: 3
  failedJobsHistoryLimit: 1
  jobTemplate:
    spec:
      completions: 4
      parallelism: 2
      template:
        <pod template>
```

```
apiVersion: batch/v1
kind: Job
metadata:
  name: job-example
spec:
  backoffLimit: 4
  completions: 4
  parallelism: 2
  template:
    spec:
      containers:
      - name: hello
        image: alpine:latest
        command: ["/bin/sh", "-c"]
        args: ["echo hello from $HOSTNAME!"]
      restartPolicy: Never
```

THANKS!

Any questions?