kubernetes

JOBS & CRONJOB

❖ WORKLOAD RESOURCES:

- **Jobs** and **CronJobs,** which are essential for running one-off tasks and scheduled tasks within a Kubernetes cluster.

➢ 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. As pods successfully complete, the Job tracks the successful completions.
- Jobs are ideal for tasks that need to be run to completion, such as batch processing, data migration, or any task that needs to be executed once.

KEY FEATURES OF JOBS:

TASK COMPLETION: A Job ensures that a specified number of pods successfully terminate.

POD RESTART POLICY: If a pod fails, the Job controller can create new pods to replace the failed ones until the task is completed.

CREATE A JOB:

```
apiVersion: batch/v1
kind: Job
metadata:
  name: hello-world
spec:
  template:
    spec:
      containers:
      - name: busybox
        image: busybox
        command: ["echo", "Hello World…!"]
      restartPolicy: Never
```

→ **To apply a Job:**

$kubectl apply -f <file-name>

$kubectl get jobs

$kubectl describe job job-name

$kubectl get pods

$kubectl logs $pods

$kubectl logs <pod-name>

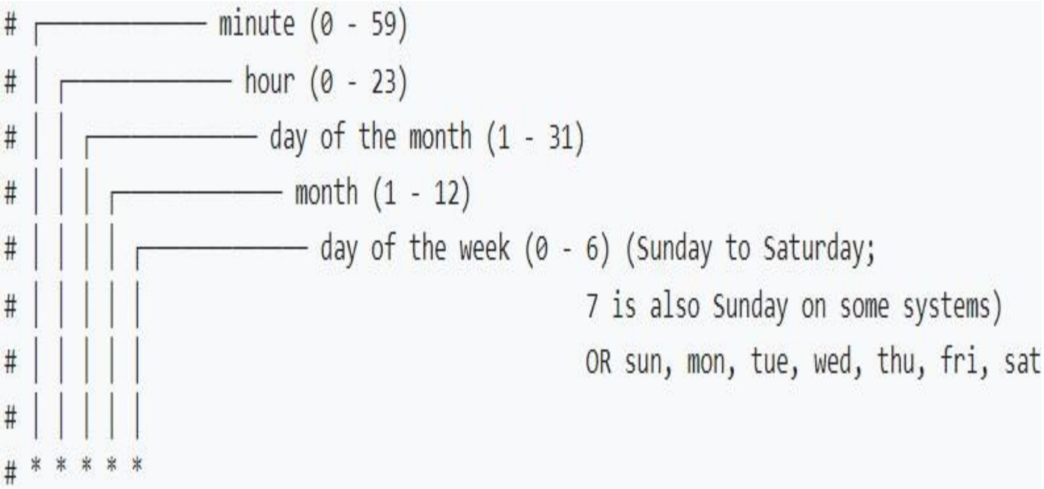
→ **To Deleting a Job:**

$kubectl delete job job-name

$kubectl get jobs

➢ CRONJOB:

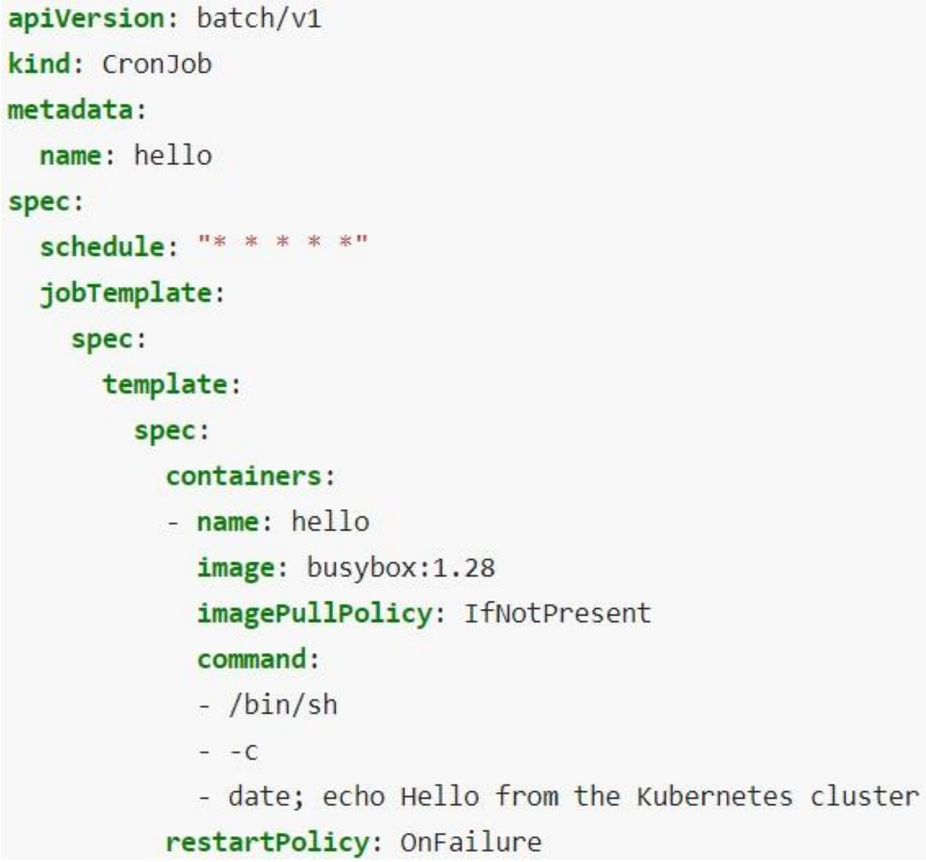
- Kubernetes CronJobs are a way to run a task on a time-based schedule and have been around for a long time in Linux and UNIX systems.
- They can be used in to run **recurring tasks** such as backup jobs, triggering emails, report generation, or automating restarts of containers.
- It runs a job **periodically** on a given schedule, written in Crontab format.



```
# ┌───────────── minute (0 - 59)
# │ ┌───────────── hour (0 - 23)
# │ │ ┌───────────── day of the month (1 - 31)
# │ │ │ ┌───────────── month (1 - 12)
# │ │ │ │ ┌───────────── day of the week (0 - 6) (Sunday to Saturday;
# │ │ │ │ │                                   7 is also Sunday on some systems)
# │ │ │ │ │                                   OR sun, mon, tue, wed, thu, fri, sat
# │ │ │ │ │
# * * * * *
```

EXAMPLE: 0 0 13 * 5 states that the task must be started every Friday at midnight, as well as on the 13th of each month at midnight.

CREATE A CRONJOB TO PRINT CURRENT TIME & MSG EVERY MINUTE:



```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "* * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox:1.28
            imagePullPolicy: IfNotPresent
            command:
            - /bin/sh
            - -c
            - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
```

→ **Create a CronJob:**

$kubectl create -f <file-name>

$kubectl get cronjobs

$kubectl describe cronjobs cronjob-name

→ **To view the running jobs in real-time using the --watch argument:**

$kubectl get jobs --watch

→ **To view the pods that have been created to run the jobs:**

$kubectl get pods

→ **To view logs from the pod to verify the command ran successfully:**

$kubectl logs hello-27827258--1-rdf4s

→ **Delete a CronJob:**

$kubectl delete cronjob cronjob-name