## Exercise 4.2: Designing Applications With Duration: Create a Job

While most applications are deployed such that they continue to be available there are some which we may want to run a particular number of times called a `Job`, and others on a regular basis called a `CronJob`

1. Create a job which will run a container which sleeps for three seconds then stops.

```
student@cp:~$ vim job.yaml
```

**job.yaml**

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: sleepy
spec:
  template:
    spec:
      containers:
      - name: resting
        image: busybox
        command: ["/bin/sleep"]
        args: ["3"]
      restartPolicy: Never
```

2. Create the job, then verify and view the details. The example shows checking the job three seconds in and then again after it has completed. You may see different output depending on how fast you type.

```
student@cp:~$ kubectl create -f job.yaml
```

```
job.batch/sleepy created
```

```
student@cp:~$ kubectl get job
```

```
NAME     COMPLETIONS   DURATION   AGE
sleepy   0/1           3s         3s
```

```
student@cp:~$ kubectl describe jobs.batch sleepy
```

```
Name:           sleepy
Namespace:      default
Selector:       controller-uid=24c91245-d0fb-11e8-947a-42010a800002
Labels:         controller-uid=24c91245-d0fb-11e8-947a-42010a800002
                job-name=sleepy
Annotations:    <none>
Parallelism:    1
Completions:    1
Start Time:     Sun, 03 Nov 2019  04:22:50 +0000
Completed At:   Sun, 03 Nov 2019  04:22:55 +0000
Duration:       5s
Pods Statuses:  0 Running / 1 Succeeded / 0 Failed
<output_omitted>
```

```
student@cp:˜$ kubectl get job
```

```
NAME     COMPLETIONS   DURATION   AGE
sleepy   1/1           5s         17s
```

3. View the configuration information of the job. There are three parameters we can use to affect how the job runs. Use `-o yaml` to see these parameters. We can see that `backoffLimit`, `completions`, and the `parallelism`. We'll add these parameters next.

```
student@cp:˜$ kubectl get jobs.batch sleepy -o yaml
```

```
<output_omitted>
  uid: c2c3a80d-d0fc-11e8-947a-42010a800002
spec:
  backoffLimit: 6
  completions: 1
  parallelism: 1
  selector:
    matchLabels:
<output_omitted>
```

4. As the job continues to `AGE` in a completion state, delete the job.

```
student@cp:˜$ kubectl delete jobs.batch sleepy
```

```
job.batch "sleepy" deleted
```

5. Edit the YAML and add the `completions:` parameter and set it to `5`.

```
student@cp:˜$ vim job.yaml
```

job.yaml

```yaml
<output_omitted>
metadata:
  name: sleepy
spec:
  completions: 5   #<--Add this line
  template:
    spec:
      containers:
<output_omitted>
```

6. Create the job again. As you view the job note that `COMPLETIONS` begins as zero of `5`.

```
student@cp:˜$ kubectl create -f job.yaml
```

```
job.batch/sleepy created
```

```
student@cp:˜$ kubectl get jobs.batch
```

```
NAME     COMPLETIONS   DURATION   AGE
sleepy   0/5           5s         5s
```

7. View the pods that running. Again the output may be different depending on the speed of typing.

```
student@cp:˜$ kubectl get pods
```

```
NAME                         READY   STATUS          RESTARTS    AGE
nginx-67f8fb575f-g4468       1/1     Running         2           2d
registry-56cffc98d6-xlhhf    1/1     Running         1           2d
sleepy-z5tnh                 0/1     Completed       0           8s
sleepy-zd692                 1/1     Running         0           3s
<output_omitted>
```

8. Eventually all the jobs will have completed. Verify then delete the job.

```
student@cp:~$ kubectl get jobs
```

```
NAME      COMPLETIONS    DURATION    AGE
sleepy    5/5            26s         10m
```

```
student@cp:~$ kubectl delete jobs.batch sleepy
```

```
job.batch "sleepy" deleted
```

9. Edit the YAML again. This time add in the `parallelism:` parameter. Set it to `2` such that two pods at a time will be deployed.

```
student@cp:~$ vim job.yaml
```

job.yaml

```yaml
<output_omitted>
  name: sleepy
spec:
  completions: 5
  parallelism: 2   #<-- Add this line
  template:
    spec:
<output_omitted>
```

10. Create the `job` again. You should see the pods deployed two at a time until all five have completed.

```
student@cp:~$ kubectl create -f job.yaml
```

```
student@cp:~$ kubectl get pods
```

```
NAME                         READY   STATUS      RESTARTS    AGE
nginx-67f8fb575f-g4468       1/1     Running     2           2d
registry-56cffc98d6-xlhhf    1/1     Running     1           2d
sleepy-8xwpc                 1/1     Running     0           5s
sleepy-xjqnf                 1/1     Running     0           5s
try1-c9cb54f5d-b45gl         2/2     Running     0           8h
<output_omitted>
```

```
student@cp:~$ kubectl get jobs
```

```
NAME      COMPLETIONS    DURATION    AGE
sleepy    3/5            11s         11s
```

11. Add a parameter which will stop the job after a certain number of seconds. Set the `activeDeadlineSeconds:` to `15`. The job and all pods will end once it runs for 15 seconds.

```
student@cp:~$ vim job.yaml
```

job.yaml

```yaml
<output_omitted>
  completions: 5
  parallelism: 2
  activeDeadlineSeconds: 15    #<-- Add this line
  template:
    spec:
      containers:
      - name: resting
        image: busybox
        command: ["/bin/sleep"]
        args: ["3"]
<output_omitted>
```

12. Delete and recreate the job again. It should run for four times then continue to age without further completions.

```
student@cp:˜$ kubectl delete jobs.batch sleepy
```

```
job.batch "sleepy" deleted
```

```
student@cp:˜$ kubectl create -f job.yaml
```

```
job.batch/sleepy created
```

```
student@cp:˜$ kubectl get jobs
```

```
NAME     COMPLETIONS   DURATION   AGE
sleepy   2/5           6s         6s
```

```
student@cp:˜$ kubectl get jobs
```

```
NAME     COMPLETIONS   DURATION   AGE
sleepy   4/5           16s        16s
```

13. View the `message:` entry in the `Status` section of the object YAML output. You may see less `status` if the job has yet to run. Wait and try again, if so.

```
student@cp:˜$ kubectl get job sleepy -o yaml
```

```
<output_omitted>
status:
  conditions:
  - lastProbeTime: "2019-11-03T16:06:10Z"
    lastTransitionTime: "2019-11-03T16:06:10Z"
    message: Job was active longer than specified deadline
    reason: DeadlineExceeded
    status: "True"
    type: Failed
  failed: 1
  startTime: "2019-11-03T16:05:55Z"
  succeeded: 4
```

14. Delete the job.

```
student@cp:˜$ kubectl delete jobs.batch sleepy
```

```
job.batch "sleepy" deleted
```