

Running Jobs and CronJobs

Relevant Documentation

- [Jobs](#)
- [CronJob](#)

Exam Tips

- A Job is designed to run a containerized task successfully to completion.
- CronJobs run Jobs periodically according to a schedule.
- The `restartPolicy` for a Job or CronJob Pod must be `OnFailure` or `Never`.
- Use `activeDeadlineSeconds` in the Job spec to terminate the Job if it runs too long.

Lesson Reference

Log in to the **control plane node**.

Create a simple Job that runs the command `echo This is a test!`.

```
vi my-job.yml
```

```
apiVersion: batch/v1
kind: Job
metadata:
  name: my-job
spec:
  template:
    spec:
      containers:
        - name: print
          image: busybox:stable
          command: ["echo", "This is a test!"]
          restartPolicy: Never
      backoffLimit: 4
      activeDeadlineSeconds: 10
```

Run the Job.

```
kubectl apply -f my-job.yml
```

Check the status of the Job.

```
kubectl get jobs
```

View the Job output. First, you will need to find the name of the Job's Pod.

```
kubectl get pods  
kubectl logs $JOB_POD_NAME
```

Create a CronJob that will run the Job task every minute.

```
vi my-cronjob.yml
```

```
apiVersion: batch/v1  
kind: CronJob  
metadata:  
  name: my-cronjob  
spec:  
  schedule: "*/1 * * * *"  
  jobTemplate:  
    spec:  
      template:  
        spec:  
          containers:  
            - name: print  
              image: busybox:stable  
              command: ["echo", "This is a test!"]  
              restartPolicy: Never  
          backoffLimit: 4  
          activeDeadlineSeconds: 10
```

```
kubectl apply -f my-cronjob.yml
```

Check the CronJob status.

```
kubectl get cronjob
```