

**Lab Manual- Kubernetes Jobs and Cron Job**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn916

**Contributor:**

Contents

[1. Objective 3](#_Toc158377332)

[2. JOB and CRONJOB 3](#_Toc158377333)

[3. Problems with Pods (Why Use ReplicaSets instead?) 4](#_Toc158377334)

[4. Kubernetes Job - Create Job 4](#_Toc158377335)

[5. Kubernetes Job - What Happen when POD deleted before Job Complete 6](#_Toc158377336)

[6. Kubernetes Job - Completion and Parallelism 8](#_Toc158377337)

[1. Completion Demo: 8](#_Toc158377338)

[1. Completion and Parallelism Demo: 9](#_Toc158377339)

[7. Kubernetes Job - Backoff Limit 11](#_Toc158377340)

[1. First Fail the POD Creation by specified wrong directory 11](#_Toc158377341)

[2. Use the Backofflimit 13](#_Toc158377342)

[8. Kubernetes Job - activeDeadlineSeconds 15](#_Toc158377343)

[9. Kubernetes Cron Job - Create Cronjob 17](#_Toc158377344)

[10. Kubernetes Cron Job - Suspend Cronjob 18](#_Toc158377345)

# Objective

<https://github.com/bipeensinha/aks-cron>

# JOB and CRONJOB

# Kubernetes Job - Create Job

Open Terminal 1 and run this command

watch kubectl get all



Open Terminal 2 and run this command

cat jobdemo.yaml

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  template:

    spec:

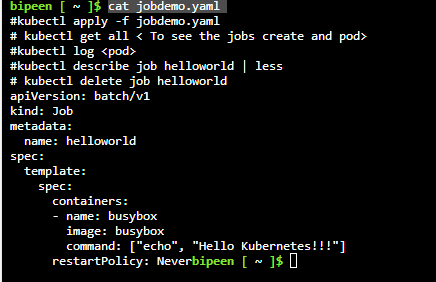
      containers:

      - name: busybox

        image: busybox

        command: ["echo", "Hello Kubernetes!!!"]

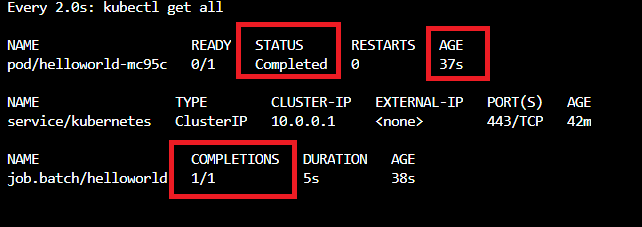
      restartPolicy: Never



kubectl apply -f jobdemo.yaml



Watch the output on Terminal 1



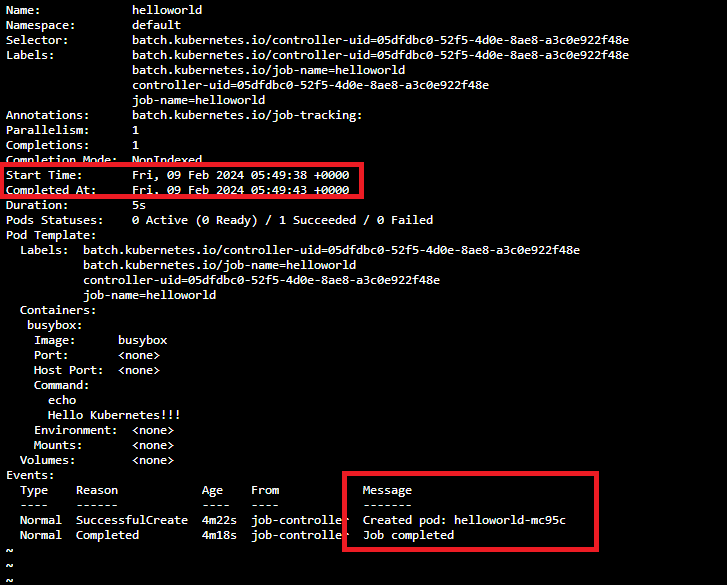
Now on Terminal 2 run below command

kubectl logs helloworld-mc95c

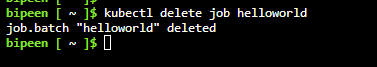


kubectl describe job helloworld | less





kubectl delete job helloworld



# Kubernetes Job - What Happen when POD deleted before Job Complete

The Job object will start a new Pod if the first Pod fails or is deleted (for example due to a node hardware failure or a node reboot).

cat jobdemo1.yaml

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  template:

    spec:

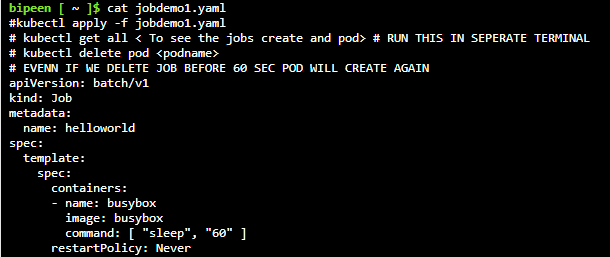
      containers:

      - name: busybox

        image: busybox

        command: [ "sleep", "60" ]

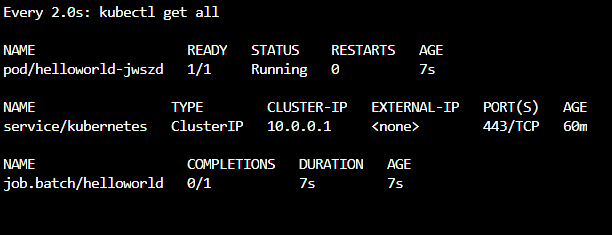
      restartPolicy: Never



kubectl apply -f jobdemo1.yaml

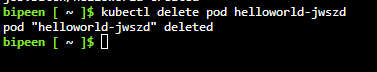


Now Watch on terminal one and copy the pod name

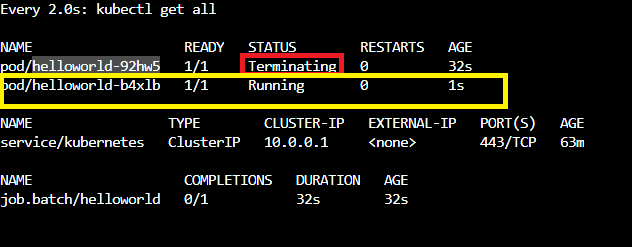


Delete the pod before 60sec

kubectl delete pod helloworld-jwszd



Now Watch on terminal 1 and you should able to see the new pod is getting created



kubectl delete job helloworld



# Kubernetes Job - Completion and Parallelism

## Completion Demo:

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  completions: 2

  template:

    spec:

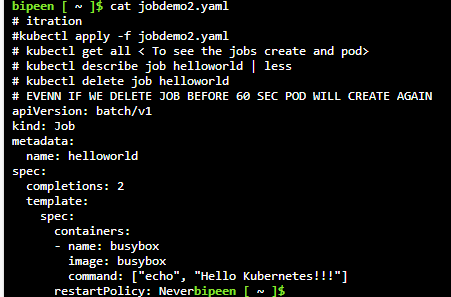
      containers:

      - name: busybox

        image: busybox

        command: ["echo", "Hello Kubernetes!!!"]

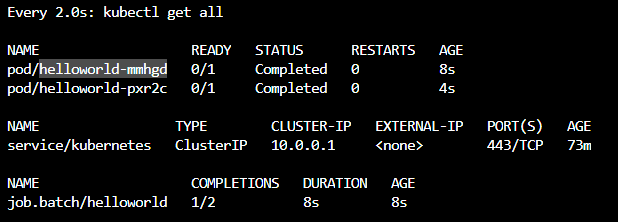
      restartPolicy: Never



kubectl apply -f jobdemo2.yaml

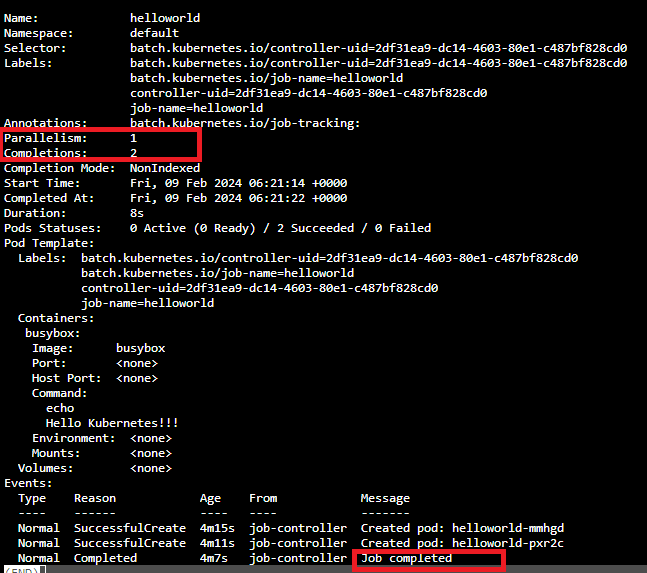


Terminal 1 watch the output



kubectl describe job helloworld | less





kubectl delete job helloworld



## Completion and Parallelism Demo:

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  completions: 2

  parallelism: 2

  template:

    spec:

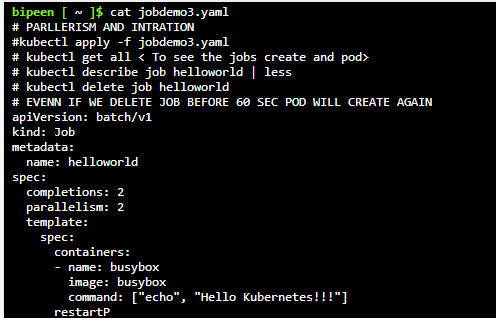
      containers:

      - name: busybox

        image: busybox

        command: ["echo", "Hello Kubernetes!!!"]

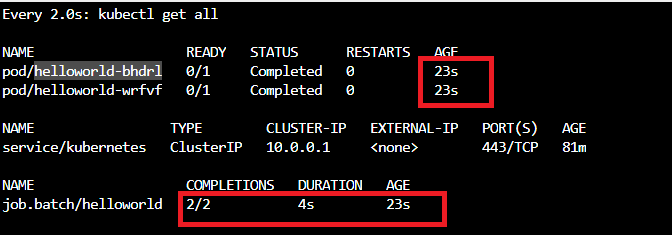
      restartPolicy: Never



kubectl apply -f jobdemo3.yaml

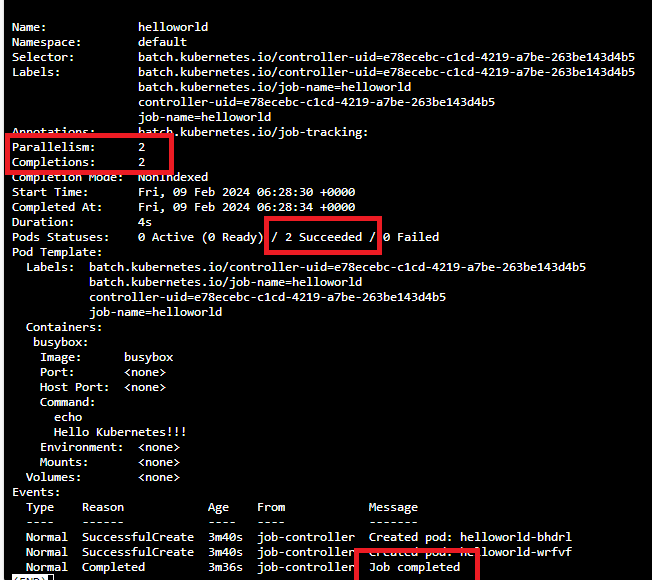


Watch on Terminal 1 both job created at same time



kubectl describe job helloworld | less





kubectl delete job helloworld



# Kubernetes Job - Backoff Limit

Kubernetes will automatically retry Jobs up to a specified number of attempts (six by default). This value is called the back-off limit

## First Fail the POD Creation by specified wrong directory

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  template:

    spec:

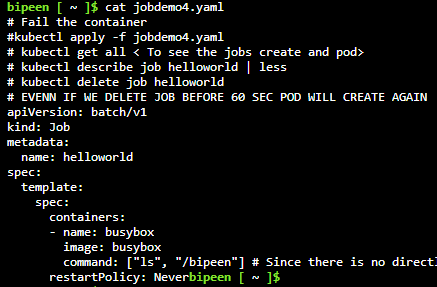
      containers:

      - name: busybox

        image: busybox

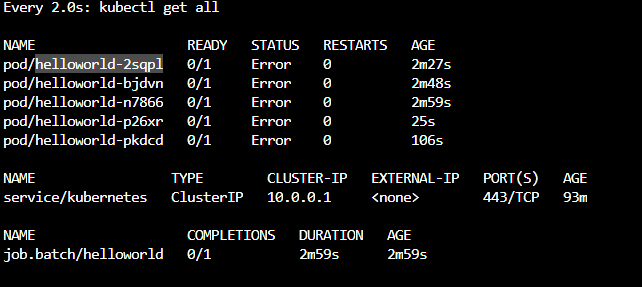
        command: ["ls", "/bipeen"] # Since there is no directlry name bipeen in Busybox container with  Fail and exit with status 0

      restartPolicy: Never



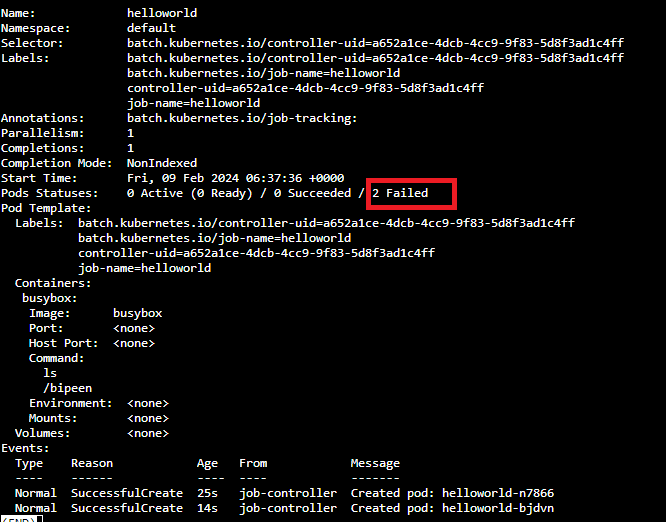
kubectl apply -f jobdemo4.yaml



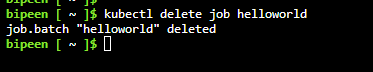


kubectl describe job helloworld | less





kubectl delete job helloworld



## Use the Backofflimit

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  backoffLimit: 2

  template:

    spec:

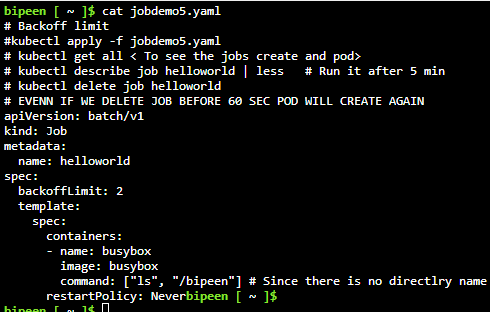
      containers:

      - name: busybox

        image: busybox

        command: ["ls", "/bipeen"] # Since there is no directlry name bipeen in Busybox container with  Fail and exit with status 0

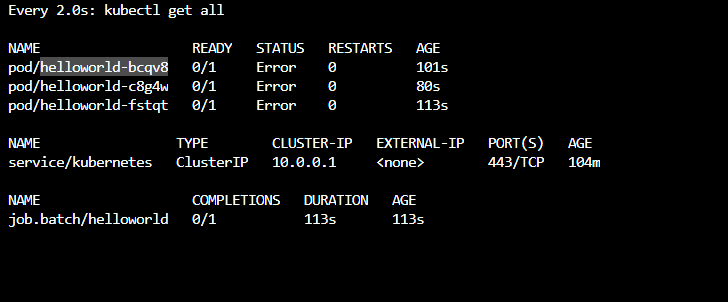
      restartPolicy: Never



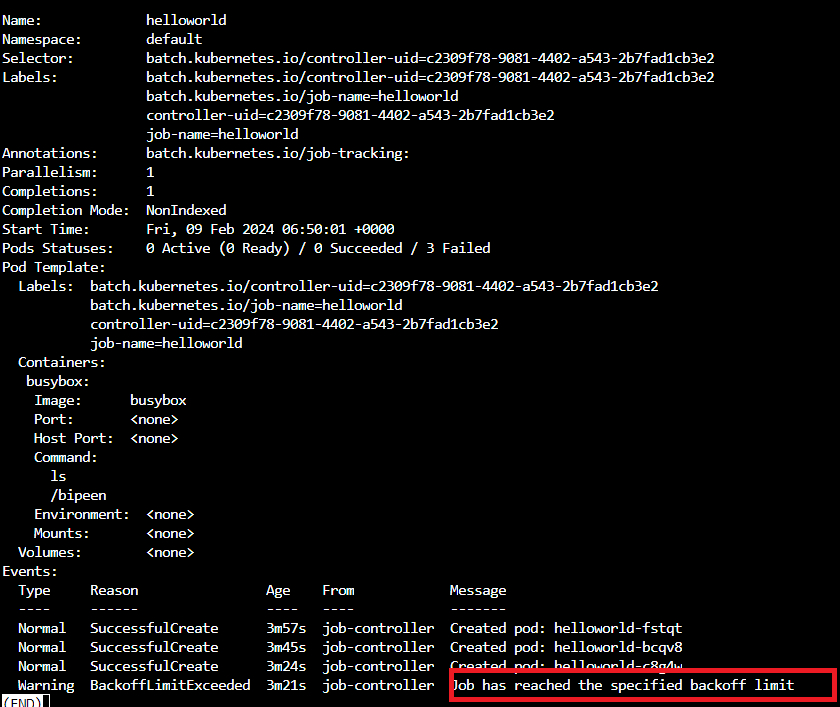
kubectl apply -f jobdemo5.yaml



After 2 Consecutive Failure , it reach the backoff limit and so after two consecutive failure it just try retry once and then it stop



kubectl describe job helloworld | less



kubectl delete job helloworld



# Kubernetes Job - activeDeadlineSeconds

Setting this parameter in the job spec ensures that the job fails if the pods it creates do not complete within the time limit specified

apiVersion: batch/v1

kind: Job

metadata:

  name: helloworld

spec:

  activeDeadlineSeconds: 10 # in sec

  template:

    spec:

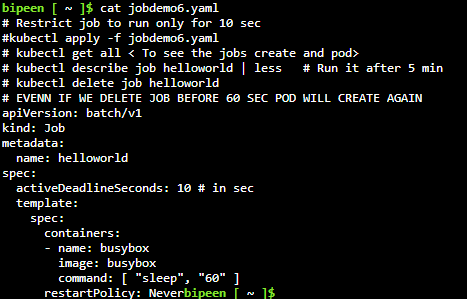
      containers:

      - name: busybox

        image: busybox

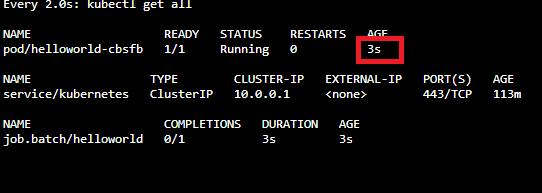
        command: [ "sleep", "60" ]

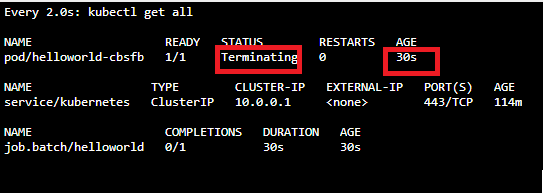
      restartPolicy: Never



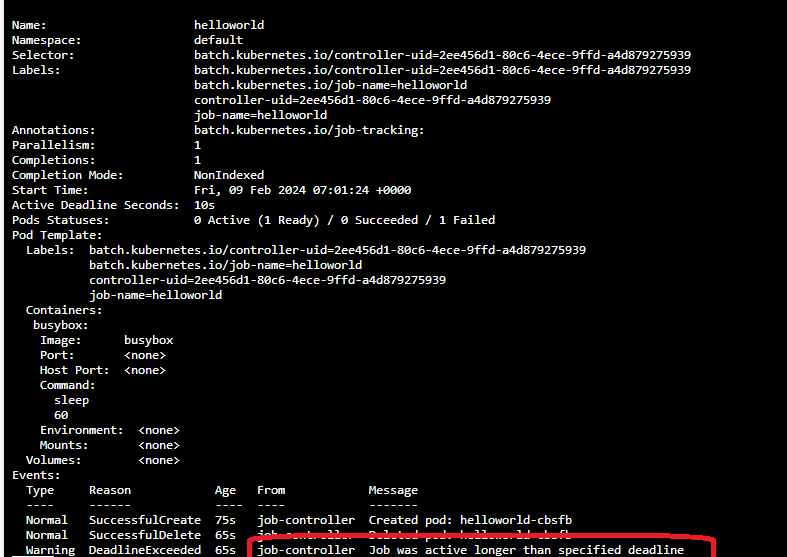
kubectl apply -f jobdemo6.yaml







kubectl describe job helloworld | less



kubectl delete job helloworld



# Kubernetes Cron Job - Create Cronjob

kubectl api-versions | grep batch



apiVersion: batch/v1

kind: CronJob

metadata:

  name: helloworld-cron

spec:

  schedule: "\* \* \* \* \*"

  jobTemplate:

    spec:

      template:

        spec:

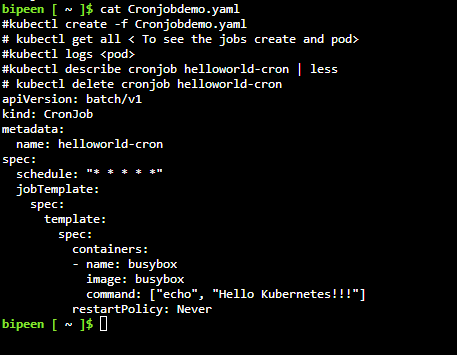
          containers:

          - name: busybox

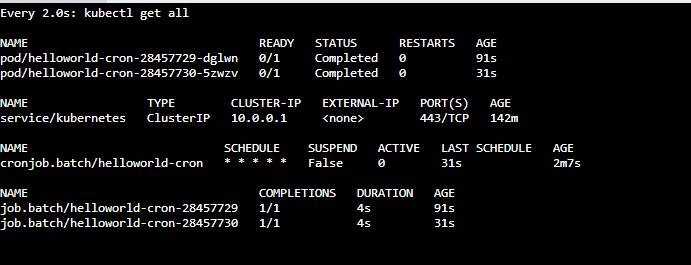
            image: busybox

            command: ["echo", "Hello Kubernetes!!!"]

          restartPolicy: Never



It keep creating the pod

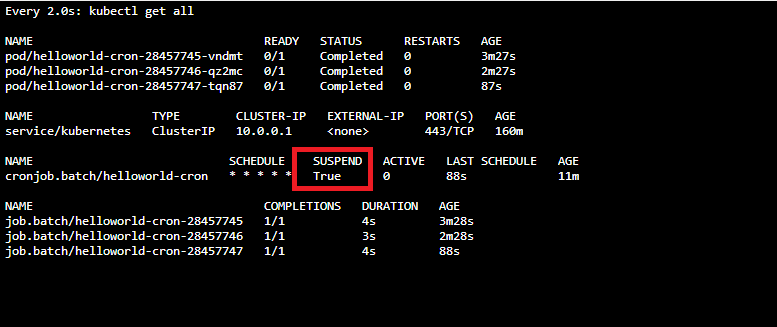


# Kubernetes Cron Job - Suspend Cronjob

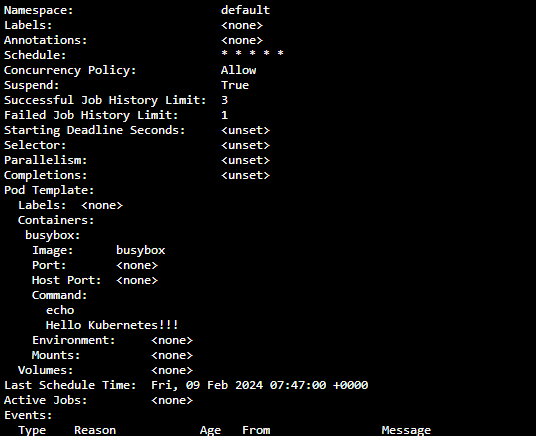
Option1 :

kubectl patch cronjob helloworld-cron -p '{"spec":{"suspend":true}}'





kubectl patch cronjob helloworld-cron -p '{"spec":{"suspend":true}}'



Option 2:

Update the specification in same yaml file



kubectl apply -f Cronjobdemo.yaml

az group delete --name AKSDemo --yes --no-wait

