JOBS

About Jobs

one-time activity is called jobs and continues activity in certain period with repetitive jobs is called cronjobs. in Kubernetes jobs and cronjobs will use for backup, restart or any command execute. in jobs is very important to verify the number of completion.

##Create a jobs##

#kubectl get jobs

```
[root@ansikube manifest]# kubectl get jobs
No resources found in default namespace.
```

cat jobs.yml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: jobs
spec:
  template:
    spec:
     containers:
       - name: pi
         image: perl
         command: ["echo", "Hello World !!!!!"]
     restartPolicy: Never
```

kubectl apply -f jobs.yml

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME          READY   STATUS      RESTARTS   AGE
jobs-99dz8    0/1     Completed   0          5s
```

@@Open another terminal and run below watch command.

#watch kubectl get all

```
Every 2.0s: kubectl get all

NAME             READY   STATUS      RESTARTS   AGE
pod/jobs-99dz8   0/1     Completed   0          32s
pod/task-pv-pod  1/1     Running     0          34m
pod/test-pd3     1/1     Running     0          65m
pod/test-pod     1/1     Running     0          96m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   9h

NAME            COMPLETIONS   DURATION   AGE
job.batch/jobs  1/1           2s         32s
```

kubectl describe jobs jobs

```
    Command:
      echo
      Hello World !!!!!
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age   From            Message
  ----    ------            ----  ----            -------
  Normal  SuccessfulCreate  78s   job-controller  Created pod: jobs-99dz8
```

@@Testing with wrong command@@

cat wrong_jobs.yml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: jobs
spec:
  template:
    spec:
     containers:
       - name: pi
         image: perl
         command: ["eeeecho", "Hello World !!!!!"]
     restartPolicy: Never
```

kubectl apply -f wrong_jobs.yml

kubectl get pods

```
Every 2.0s: kubectl get all

NAME              READY   STATUS              RESTARTS   AGE
pod/jobs-4c4nv    0/1     ContainerCannotRun  0          29s
pod/jobs-9cwz6    0/1     ContainerCannotRun  0          27s
pod/jobs-hcltz    0/1     ContainerCannotRun  0          17s
pod/task-pv-pod   1/1     Running             0          43m
pod/test-pd3      1/1     Running             0          73m
pod/test-pod      1/1     Running             0          105m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   9h

NAME             COMPLETIONS   DURATION   AGE
job.batch/jobs   0/1           29s        29s
```

Note:- If command is incorrect then jobs will be failed.

##Create a job with backoff limits.

cat jobs-backlimit.yml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: jobs-1
spec:
  template:
    spec:
      containers:
        - name:  pi
          image: perl
          command: ["eeecho", "Hello World !!!!!"]
      restartPolicy: Never
  backoffLimit: 4
```

kubectl apply -f jobs-backlimit.yml

kubectl get jobs

```
[root@ansikube manifest]# kubectl get jobs
NAME     COMPLETIONS   DURATION   AGE
jobs-1   0/1           17s        17s
```

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME           READY   STATUS              RESTARTS   AGE
jobs-1-8fb4b   0/1     ContainerCannotRun  0          50s
jobs-1-cknz4   0/1     ContainerCannotRun  0          80s
jobs-1-glct2   0/1     ContainerCannotRun  0          10s
jobs-1-qh7pd   0/1     ContainerCannotRun  0          82s
jobs-1-vz6h2   0/1     ContainerCannotRun  0          70s
```

@@Open another terminal and run below watch command.

watch kubectl get all

```
Every 2.0s: kubectl get all

NAME              READY   STATUS              RESTARTS   AGE
pod/jobs-1-8fb4b  0/1     ContainerCannotRun  0          4m47s
pod/jobs-1-cknz4  0/1     ContainerCannotRun  0          5m17s
pod/jobs-1-glct2  0/1     ContainerCannotRun  0          4m7s
pod/jobs-1-qh7pd  0/1     ContainerCannotRun  0          5m19s
pod/jobs-1-vz6h2  0/1     ContainerCannotRun  0          5m7s
pod/task-pv-pod   1/1     Running             0          3h53m
pod/test-pd3      1/1     Running             0          4h23m
pod/test-pod      1/1     Running             0          4h55m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   13h

NAME              COMPLETIONS   DURATION   AGE
job.batch/jobs-1  0/1           5m19s      5m19s
```

kubectl describe job jobs-1

```
Events:
  Type     Reason                Age    From            Message
  ----     ------                ----   ----            -------
  Normal   SuccessfulCreate      7m46s  job-controller  Created pod: jobs-1-qh7pd
  Normal   SuccessfulCreate      7m44s  job-controller  Created pod: jobs-1-cknz4
  Normal   SuccessfulCreate      7m34s  job-controller  Created pod: jobs-1-vz6h2
  Normal   SuccessfulCreate      7m14s  job-controller  Created pod: jobs-1-8fb4b
  Normal   SuccessfulCreate      6m34s  job-controller  Created pod: jobs-1-glct2
  Warning  BackoffLimitExceeded  5m14s  job-controller  Job has reached the specified backoff limit
[root@ansikube manifest]#
```

Note:- backoff limit has set for 4, so after 5th it will stop trying the jobs.

##Create a job with Completions, means after completing one job only another job will run.

cat jobs-1.yml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: jobs-2
spec:
  template:
    spec:
      containers:
        - name:  pi
          image: perl
          command: ["echo", "Hello World !!!!!"]
      restartPolicy: Never
  backoffLimit: 4
  completions: 2
```

kubectl apply -f jobs-1.yml

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
jobs-2-lvq6l   0/1     Completed   0          51s
jobs-2-pdk9g   0/1     Completed   0          54s
```

#watch kubectl get all

```
Every 2.0s: kubectl get all

NAME              READY   STATUS      RESTARTS   AGE
pod/jobs-2-lvq6l  0/1     Completed   0          4m22s
pod/jobs-2-pdk9g  0/1     Completed   0          4m25s
pod/task-pv-pod   1/1     Running     0          4h8m
pod/test-pd3      1/1     Running     0          4h38m
pod/test-pod      1/1     Running     0          5h10m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   13h

NAME              COMPLETIONS   DURATION   AGE
job.batch/jobs-2  2/2           5s         4m25s
```

kubectl describe jobs jobs-2

```
Events:
  Type    Reason            Age    From            Message
  ----    ------            ----   ----            -------
  Normal  SuccessfulCreate  7m53s  job-controller  Created pod: jobs-2-pdk9g
  Normal  SuccessfulCreate  7m50s  job-controller  Created pod: jobs-2-lvq6l
```

Note:- completions: 2 has mentioned, so it will get completed with 2 jobs.

##Create a job by increasing Parallelism. At the same time two pods will be created.

cat jobs-2.yml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: jobs-2
spec:
  template:
    spec:
      containers:
        - name:  pi
          image: perl
          command: ["echo", "Hello World !!!!!"]
      restartPolicy: Never
  backoffLimit: 4
  parallelism: 2
  completions: 2
```

kubectl apply -f jobs-2.yml

kubectl get jobs

```
[root@ansikube manifest]# kubectl get jobs
NAME     COMPLETIONS   DURATION   AGE
jobs-2   2/2           5s         21m
[root@ansikube manifest]#
```

@@Open another terminal and run below watch command.

watch kubectl get all

```
Every 2.0s: kubectl get all

NAME              READY   STATUS      RESTARTS   AGE
pod/jobs-2-lvq6l  0/1     Completed   0          23m
pod/jobs-2-pdk9g  0/1     Completed   0          23m
pod/task-pv-pod   1/1     Running     0          4h27m
pod/test-pd3      1/1     Running     0          4h57m
pod/test-pod      1/1     Running     0          5h29m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   13h

NAME              COMPLETIONS   DURATION   AGE
job.batch/jobs-2  2/2           5s         23m
```

kubectl describe jobs jobs-2

```
Parallelism:    2
Completions:    2
Start Time:     Wed, 06 Nov 2019 13:16:38 +0000
Completed At:   Wed, 06 Nov 2019 13:16:43 +0000
Duration:       5s
Pods Statuses:  0 Running / 2 Succeeded / 0 Failed
```

Note:- Parallelism 2 has given, so at the same time jobs will be executed.

##Create a cronjobs## By adding schedule

cat cronjob.yml

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox
            args:
            - /bin/sh
            - -c
            - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
```

kubectl apply -f cronjob.yml

kubectl get jobs

```
[root@ansikube manifest]# kubectl get jobs
NAME      COMPLETIONS   DURATION   AGE
jobs-2    2/2           5s         35m
[root@ansikube manifest]#
```

@@Open another terminal and run below watch command.

watch kubectl get all

```
Every 2.0s: kubectl get all

NAME                         READY   STATUS      RESTARTS   AGE
pod/hello-1573048320-bqhqn   0/1     Completed   0          101s
pod/hello-1573048380-dc57j   0/1     Completed   0          41s
pod/jobs-2-lvq6l             0/1     Completed   0          37m
pod/jobs-2-pdk9g             0/1     Completed   0          37m
pod/task-pv-pod              1/1     Running     0          4h40m
pod/test-pd3                 1/1     Running     0          5h11m
pod/test-pod                 1/1     Running     0          5h42m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.36.0.1    <none>        443/TCP   13h

NAME                         COMPLETIONS   DURATION   AGE
job.batch/hello-1573048320   1/1           2s         101s
job.batch/hello-1573048380   1/1           2s         41s
job.batch/jobs-2             2/2           5s         37m

NAME                SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
cronjob.batch/hello   */1 * * * *   False     0        41s             118s
```

kubectl get jobs

```
[root@ansikube manifest]# kubectl get jobs
NAME               COMPLETIONS   DURATION   AGE
hello-1573048320   1/1           2s         2m37s
hello-1573048380   1/1           2s         97s
hello-1573048440   1/1           2s         37s
```

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME                     READY   STATUS      RESTARTS   AGE
hello-1573048380-dc57j   0/1     Completed   0          2m10s
hello-1573048440-fmwwc   0/1     Completed   0          70s
hello-1573048500-t2r6v   0/1     Completed   0          10s
```

Note:- every one-minute jobs will be run.