

## 4.5 완료가능한 단일 태스크를 수행하는 파드 실행

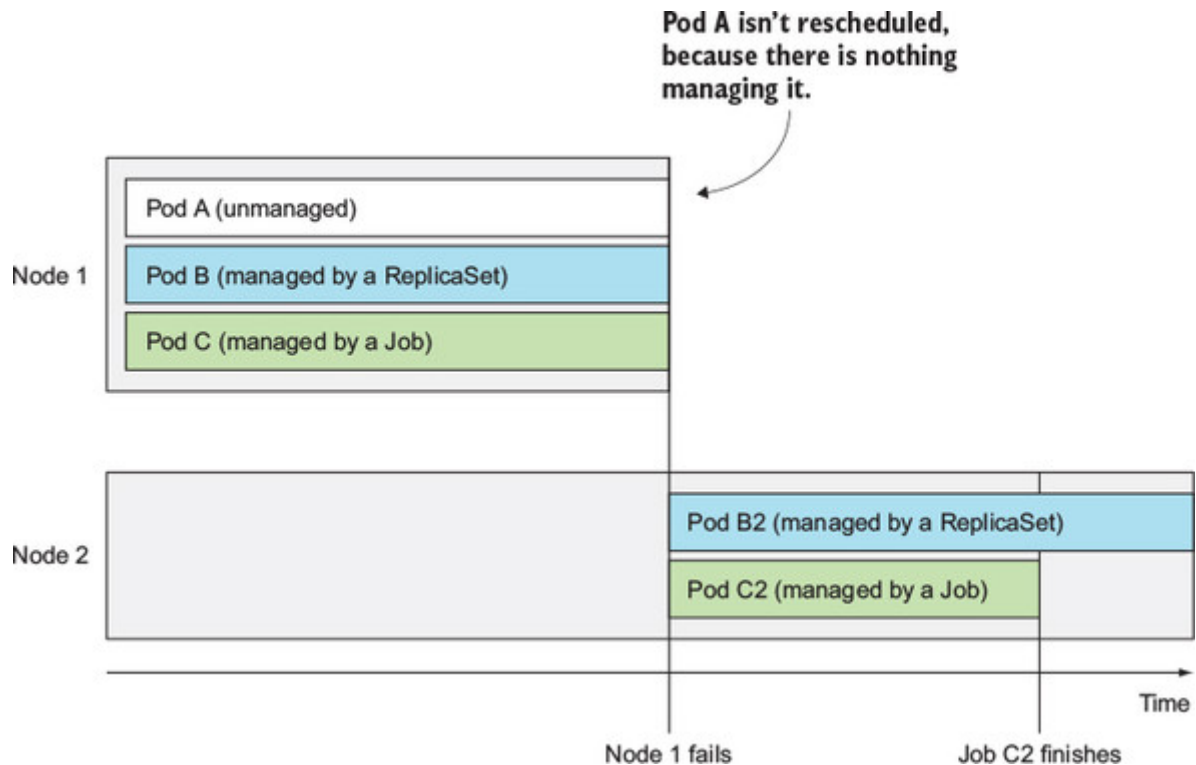
🕒 생성일	@2021년 3월 27일 오후 2:39
🏷 태그	

작업 완료 후, 종료되는 태스크만 실행하려는 경우?!  
앞에 RC, RS, DS는 완료됐다고 간주되지 않는 지속적인 태스크 실행.  
→ 그렇다면 완료가능한 태스크에서는 프로세스가 종료된 후 다시 시작되지 않음

### 1. 잡 리소스 소개

잡(Job) : 계속해서 실행되어야하는 성격이 아니라 **실행되고나서 종료되어야하는 성격의 작업을 실행시킬때** 사용

- 특정 개수 만큼의 파드가 성공적으로 완료되는걸 보장해줌
- 잡이 파드 하나 실행 → 파드 정상 종료되었는지 확인
  - 노드 장애 발생시 → RS 파드와 같은 방식으로 다른 노드에 리스케줄
  - 프로세스 자체에 장애가 발생한 경우 → 잡에서 컨테이너 재시작할 지 설정가능
- 제대로 작업이 완료되는 것이 중요한 임시 작업에 유용함



초기에 스케줄링된 노드 1이 장애 발생시, 잡에 의해 생성된 Pod C를 새 노드 2로 리스케줄하는 것을 볼 수 있음

## 2. 잡 리소스 정의

데이터를 어딘가에 저장하고 있고, 해당 데이터를 변환해서 전송하는 경우

→ 잡을 완료하는데 시간이 많이 걸리는 작업을 자동화하는 것을 emulation 해보자

vi로 `exporter.yaml` 매니페스트 생성

```
apiVersion: batch/v1 #batch API 그룹에 속함
kind: Job
metadata:
  name: batch-job
spec: #파드 셀렉터를 지정하지 않음 (파드 템플릿의 레이블 기반으로 생성)
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure # 잡은 기본 재시작 정책(Always)을 사용못함
      containers:
        - name: main
          image: luksa/batch-job
```

- luksa/batch-job 이미지 : 정확히 120초동안 실행된 뒤 종료되는 프로세스 호출
- restartPolicy : 기본값은 Always 지만, 잡 파드는 무한정 실행하지 않고 종료되는 성격이므로 명시적으로 OnFailure나 Never 설정(컨테이너 종료시 재시작 하지 않는것)

### 3. 파드를 실행한 잡 보기

```
kubectl create -f exporter.yaml
kubectl get jobs
kubectl get po
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get jobs
NAME          COMPLETIONS  DURATION  AGE
batch-job     0/1          27s       27s
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME          READY  STATUS   RESTARTS  AGE
batch-job-2sq5g  1/1    Running  0          37s
```

아직 러닝중이당..



```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get jobs
NAME          COMPLETIONS  DURATION  AGE
batch-job     1/1           2m5s      2m37s
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME          READY  STATUS    RESTARTS  AGE
batch-job-2sq5g  0/1    Completed  0          2m43s
```

꽤 완료된것으로 뜬당.....!<3

책에서는 DESIRED SUCCESSFUL 칼럼이 있지만, 이제 COMPLETIONS로 바뀌었다.

## 파드가 완료될 때 파드가 삭제되지 않은 이유

해당 파드의 로그를 검사할 수 있게 하기 위해서!

```
kubectl logs batch-job-2sq5g
```

이렇게 로그를 볼 수 있당.

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl logs batch-job-2sq5g
Sat Mar 27 06:00:07 UTC 2021 Batch job starting
Sat Mar 27 06:02:07 UTC 2021 Finished succesfully
```

## 4. 잡에서 여러 파드 인스턴스 실행하기

잡은 병렬, 순차적으로 실행하도록 구성할 수 있다.

다음 속성을 설정해서 수행할 수 있당

- completions
- parallelism

### 순차적으로 잡 파드 실행하기

잡을 두 번 이상 실행해야 하는 경우, 잡의 파드를 몇 번 실행할지 `completions`에 설정

vi로 여러번 완료시키는 잡 설정 `multi-completion-batch-job.yaml` 만들자

```
apiVersion: batch/v1
kind: Job
metadata:
```

```

name: multi-completion-batch-job
spec:
  completions: 5 # 5개 파드를 순차적으로 실행한당!!!
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure
      containers:
        - name: main
          image: luksa/batch-job

```

## 병렬로 잡 파드 실행하기

여러 파드를 병렬로 실행하게끔 `parallelism` 속성을 이용하자.

vi로 `multi-completion-parallel-batch-job.yaml` 맵글자

```

apiVersion: batch/v1
kind: Job
metadata:
  name: multi-completion-batch-job
spec:
  completions: 5 # 이 잡은 다섯개의 파드를 성공적으로 완료해야함
  parallelism: 2 # 두개까지 병렬로 실행 가능
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure
      containers:
        - name: main
          image: luksa/batch-job

```

2개 씩 동시에 실행되는 것을 볼 수 있다

```

jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
batch-job-2sq5g                     0/1     Completed 0           22m
multi-completion-batch-job-7jk8z    1/1     Running   0           35s
multi-completion-batch-job-hst8m    1/1     Running   0           35s

```

```

jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
batch-job-2sq5g                     0/1     Completed 0           24m
multi-completion-batch-job-7jk8z    0/1     Completed 0           2m32s
multi-completion-batch-job-hst8m    0/1     Completed 0           2m32s
multi-completion-batch-job-msgmj    1/1     Running   0           27s
multi-completion-batch-job-nfttv    1/1     Running   0           25s

```

## 잡 스케일링

잡 실행 중에 parallelism 속성 변경 가능

```
kubectl scale job multi-completion-batch-job-nfttv --replicas 3
```

이렇게 하면 동시에 3개를 병렬로 실행 가능 (parallelism을 2에서 3으로 증가시킴)

## 5. 잡 파드가 완료되는 데 걸리는 시간 제한하기

- 잡 파드가 특정 상태에 빠져서 완료하는데 너무 오래 걸리는 경우
  - 파드 스펙에 `activeDeadlineSeconds` 속성을 설정해 파드의 실행 시간을 제한 가능
  - 이것보다 오래 실행되면 시스템이 종료 시도 : 잡 상태는 실패로 표시
- 잡 재시도 횟수 설정 가능

`spec.backoffLimit` 필드를 지정 (기본값 = 6)