perform a predefined finite unit of work reliably and then shut down the container. For this task, Kubernetes provides the Job resource.

Solution

A Kubernetes Job is similar to a ReplicaSet as it creates one or more Pods and ensures they run successfully. However, the difference is that, once the expected number of Pods terminate successfully, the Job is considered complete and no additional Pods are started. A Job definition looks like Example 7-1.

Example 7-1. A Job specification

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: random-generator
spec:
  completions: 5                         ❶
  parallelism: 2                         ❷
  template:
    metadata:
      name: random-generator
    spec:
      restartPolicy: OnFailure           ❸
      containers:
      - image: k8spatterns/random-generator:1.0
        name: random-generator
        command: [ "java", "-cp", "/", "RandomRunner", "/numbers.txt", "10000" ]
```

❶ Job should run five Pods to completion, which all must succeed.

❷ Two Pods can run in parallel.

❸ Specifying the `restartPolicy` is mandatory for a Job.

One crucial difference between the Job and the ReplicaSet definition is the `.spec.template.spec.restartPolicy`. The default value for a ReplicaSet is `Always`, which makes sense for long-running processes that must always be kept running. The value `Always` is not allowed for a Job and the only possible options are either `OnFailure` or `Never`.