Controller objects to know about

This reading explains the relationship among several Kubernetes controller objects:

- ReplicaSets
- Deployments
- Replication Controllers
- StatefulSets
- DaemonSets
- Jobs

A ReplicaSet controller ensures that a population of Pods, all identical to one another, are running at the same time. Deployments let you do declarative updates to ReplicaSets and Pods. In fact, Deployments manage their own ReplicaSets to achieve the declarative goals you prescribe, so you will most commonly work with Deployment objects.

Deployments let you create, update, roll back, and scale Pods, using ReplicaSets as needed to do so. For example, when you perform a rolling upgrade of a Deployment, the Deployment object creates a second ReplicaSet, and then increases the number of Pods in the new ReplicaSet as it decreases the number of Pods in its original ReplicaSet.

Replication Controllers perform a similar role to the combination of ReplicaSets and Deployments, but their use is no longer recommended. Because Deployments provide a helpful "front end" to ReplicaSets, this training course chiefly focuses on Deployments.

If you need to deploy applications that maintain local state, StatefulSet is a better option. A StatefulSet is similar to a Deployment in that the Pods use the same container spec. The Pods created through Deployment are not given persistent identities, however; by contrast, Pods created using StatefulSet have unique persistent identities with stable network identity and persistent disk storage.

If you need to run certain Pods on all the nodes within the cluster or on a selection of nodes, use DaemonSet. DaemonSet ensures that a specific Pod is always running on all or some subset of the nodes. If new nodes are added, DaemonSet will automatically set up Pods in those nodes with the required specification. The word "daemon" is a computer science term meaning a non-interactive process that provides useful services to other processes. A Kubernetes cluster might use a DaemonSet to ensure that a logging agent like fluentd is running on all nodes in the cluster.

The Job controller creates one or more Pods required to run a task. When the task is completed, Job will then terminate all those Pods. A related controller is CronJob, which runs Pods on a time-based schedule.

Later modules in this specialization will cover these controllers in more depth.