tions and lifecycle management capabilities. Some applications need help to warm up, and some applications need a gentle and clean shutdown procedure. For this and other use cases, some events, as shown in Figure 5-1, are emitted by the platform that the container can listen to and react to if desired.

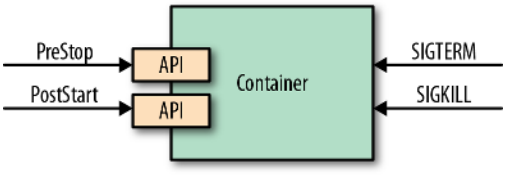

Figure 5-1. Managed container lifecycle

The deployment unit of an application is a Pod. As you already know, a Pod is composed of one or more containers. At the Pod level, there are other constructs such as init containers, which we cover in Chapter 14, *Init Container* (and defer-containers, which is still at the proposal stage as of this writing) that can help manage the container lifecycle. The events and hooks we describe in this chapter are all applied at an individual container level rather than Pod level.

SIGTERM Signal

Whenever Kubernetes decides to shut down a container, whether that is because the Pod it belongs to is shutting down or simply a failed liveness probe causes the container to be restarted, the container receives a SIGTERM signal. SIGTERM is a gentle poke for the container to shut down cleanly before Kubernetes sends a more abrupt SIGKILL signal. Once a SIGTERM signal has been received, the application should shut down as quickly as possible. For some applications, this might be a quick termination, and some other applications may have to complete their in-flight requests, release open connections, and clean up temp files, which can take a slightly longer time. In all cases, reacting to SIGTERM is the right moment to shut down a container in a clean way.

SIGKILL Signal

If a container process has not shut down after a SIGTERM signal, it is shut down forcefully by the following SIGKILL signal. Kubernetes does not send the SIGKILL signal immediately but waits for a grace period of 30 seconds by default after it has issued a SIGTERM signal. This grace period can be defined per Pod using the `.spec.terminationGracePeriodSeconds` field, but cannot be guaranteed as it can be overridden while issuing commands to Kubernetes. The aim here should be to design and implement containerized applications to be ephemeral with quick startup and shutdown processes.