Besides resource requirements, application runtimes also have dependencies on platform-managed capabilities like data storage or application configuration.

Solution

Knowing the runtime requirements for a container is important mainly for two reasons. First, with all the runtime dependencies defined and resource demands envisaged, Kubernetes can make intelligent decisions for where to place a container on the cluster for most efficient hardware utilization. In an environment with shared resources among a large number of processes with different priorities, the only way for a successful coexistence is to know the demands of every process in advance. However, intelligent placement is only one side of the coin.

The second reason container resource profiles are essential is capacity planning. Based on the particular service demands and the total number of services, we can do some capacity planning for the different environments and come up with the most cost-effective host profiles to satisfy the entire cluster demand. Service resource profiles and capacity planning go hand-to-hand for successful cluster management in the long term.

Let's have a look first at how to declare runtime dependencies before we dive into resource profiles.

Runtime Dependencies

One of the most common runtime dependencies is file storage for saving application state. Container filesystems are ephemeral and lost when a container is shut down. Kubernetes offers volume as a Pod-level storage utility that survives container restarts.

The most straightforward type of volume is `emptyDir`, which lives as long as the Pod lives and when the Pod is removed, its content is also lost. The volume needs to be backed by some other kind of storage mechanism to have a volume that survives Pod restarts. If your application needs to read or write files to such long-lived storage, you have to declare that dependency explicitly in the container definition using volumes, as shown in Example 2-1.

Example 2-1. Dependency on a PersistentVolume

```
apiVersion: v1
kind: Pod
metadata:
  name: random-generator
spec:
  containers:
  - image: k8spatterns/random-generator:1.0
```