

学习内容：

Kubernetes 教程

收获：

- emptyDir 类型的数据卷分配时是空白的，默认存储在磁盘上，也可以 emptyDir.medium 字段为"Memory"，此时 Kubernetes 将挂载一个 tmpfs（基于 RAM 的文件系统），用于实现容器之间共享内容。
- nfs 类型的数据卷可以加载 NFS（Network File System）到您的容器组/容器。容器组被移除时，将仅仅 umount（卸载）NFS 数据卷，NFS 中的数据仍将被保留。可以在加载 NFS 数据卷前就在其中准备好数据，可以在不同容器组之间共享数据，可以被多个容器组加载并同时读写。
- hostPath 类型的数据卷将 Pod（容器组）所在节点的文件系统上某一个文件或文件夹挂载进容器组（容器），依赖于宿主机。
- ConfigMap 提供了一种向容器组注入配置信息的途径。ConfigMap 中的数据可以被 Pod（容器组）中的容器作为一个数据卷挂载。
- 引用 ConfigMap 时：
 1. 可以直接引用整个 ConfigMap 到数据卷，此时 ConfigMap 中的每一个 key 对应一个文件名，value 对应文件的内容。
 2. 可以只引用 ConfigMap 中的某一个名值对，此时可以将 key 映射成一个新的文件名。
- secret 数据卷可以用来注入敏感信息（例如密码）到容器组，使用 tmpfs（基于 RAM 的文件系统）挂载。
- 将 Secret 数据卷挂载到容器时，如果该挂载点指定了数据卷内子路径，则该 Secret 被改变后，该容器挂载的内容仍然不变。
- persistentVolumeClaim 数据卷用来挂载 PersistentVolume 存储卷。
- 使用 volumeMounts.subPath 属性，可以使容器在挂载数据卷时指向数据卷内部的一个子路径，而不是直接指向数据卷的根路径。
- 使用 volumeMounts.subPathExpr 字段，可以通过容器的环境变量指定容器内路径。
- 同一个 volumeMounts 中 subPath 字段和 subPathExpr 字段不能同时使用。
- 数据卷的挂载传播由 Pod 的 spec.containers[*].volumeMounts.mountPropagation 字段控制：
 1. None：默认值。在数据卷被挂载到容器之后，此数据卷不会再接受任何后续宿主机或其他容器挂载到该数据卷对应目录下的子目录的挂载。同样的，在容器中向该数据卷对应目录挂载新目录时，宿主机也不能看到。对应 Linux 的 private mount propagation 选项 Linux 内核文档(opens new window)
 2. HostToContainer：在数据卷被挂载到容器之后，宿主机向该数据卷对应目录添加挂载时，对容器是可见的。
 3. Bidirectional：在数据卷被挂载到容器之后，宿主机向该数据卷对应目录添加挂载时，对容器是可见的；同时，从容器中向该数据卷创建挂载，同样也对宿主机可见。