**Volumes**

Kubernetes volumes by example

A Kubernetes volume is essentially a directory accessible to all containers running in a pod. In contrast to the container-local filesystem, the data in volumes is preserved across container restarts. The medium backing a volume and its contents are determined by the volume type:

* node-local types such as emptyDir or hostPath
* file-sharing types such as nfs
* cloud provider-specific types like awsElasticBlockStore, azureDisk, or gcePersistentDisk
* distributed file system types, for example glusterfs or cephfs
* special-purpose types like secret, gitRepo

A special type of volume is PersistentVolume, which we will cover elsewhere.

Let’s create a [pod](https://github.com/openshift-evangelists/kbe/blob/master/specs/volumes/pod.yaml) with two containers that use an emptyDir volume to exchange data:

$ kubectl apply -f https://raw.githubusercontent.com/openshift-evangelists/kbe/master/specs/volumes/pod.yaml

$ kubectl describe pod sharevol

Name: sharevol

Namespace: default

...

Volumes:

xchange:

Type: EmptyDir (a temporary directory that shares a pod's lifetime)

Medium:

We first exec into one of the containers in the pod, c1, check the volume mount and generate some data:

$ kubectl **exec** -it sharevol -c c1 -- bash

[root@sharevol /]# mount | grep xchange

/dev/sda1 on /tmp/xchange **type** ext4 (rw,relatime,data=ordered)

[root@sharevol /]# echo 'some data' > /tmp/xchange/data

When we now exec into c2, the second container running in the pod, we can see the volume mounted at /tmp/data and are able to read the data created in the previous step:

$ kubectl **exec** -it sharevol -c c2 -- bash

[root@sharevol /]# mount | grep /tmp/data

/dev/sda1 on /tmp/data **type** ext4 (rw,relatime,data=ordered)

[root@sharevol /]# cat /tmp/data/data

some data

Note that in each container you need to decide where to mount the volume and that for emptyDir you currently can not specify resource consumption limits.

You can remove the pod with:

$ kubectl delete pod/sharevol

As already described, this will destroy the shared volume and all its contents.