

NodePort can either be assigned randomly by K8s at service creation time, or assigned manually or programmatically.

Services can then be further exposed by using an Ingress. Ingress is not a Service type, but it acts as the entry point for a cluster. It consolidates routing rules into a single resource as it can expose multiple services under the same IP address. An ingress controller is required if you choose to expose services as Ingress.

Labels and Selectors

Labels are key/value pairs that are attached to specify identifying attributes of those objects that are meaningful and relevant to users, but do not directly imply semantics to the core system. Labels can be used to organize and to select subsets of objects. Labels can be attached to objects at creation time and subsequently added and modified at any time. Each object can have a set of key/value labels defined. Each Key must be unique for a given object. Via a *label selector*, the client/user can identify a set of objects. The label selector is the core grouping primitive in K8s.

Volume and Persistent Volume

On-disk files in a container are ephemeral (anything they store to their “local file system” cannot be guaranteed to survive a restart). This condition presents some problems for non-trivial applications that run in containers:

- ▶ When a container crashes, K8s restarts it, but the files are lost.
- ▶ When you run containers together in a pod, it is often necessary to share files between those containers.

A K8s Volume abstraction solves both problems. A volume is a piece of storage that is managed outside of the K8s clusters, and is accessible to the containers in a pod. How that storage comes to be, the medium that backs it, and the contents of it are determined by the particular volume type used.

In this publication, the types of volumes discussed are as follows:

- ▶ NFS (Network File System): An NFS volume allows an existing Network File System share to be mounted into a pod.
- ▶ RBD (Rados Block Device): An RBD volume allows a dynamically generated Ceph block device (<https://docs.ceph.com/docs/master/>) to be mounted into a pod.

For a full list of K8s volume types refer to:

<https://kubernetes.io/docs/concepts/storage/volumes/>

Additionally, volumes have different access modes, often dependent on the volume type:

- ▶ ReadWriteOnce – the volume can be mounted as read-write by a single node
- ▶ ReadOnlyMany – the volume can be mounted read-only by many nodes
- ▶ ReadWriteMany – the volume can be mounted as read/write by many nodes

When the volume becomes a resource available to and manageable by the K8s cluster, it is called a Persistent Volume (PV).

The consistence and persistence of the data on the storage layer is not K8s’ responsibility: it is typically delegated to either the storage technology, or to the containerized applications.

Persistent Volume Claim

A Persistent Volume Claim (PVC) is a request for storage by a consumer, typically a Deployment (described in “Deployment” on page 110) or a StatefulSet (described in