

Volume Drivers

Volume drivers implement the [Volume Plugin Interface](#). This provides an interface to register a volume driver and advertise the driver to Docker. Registering a driver with this volume interface will cause Docker to be able to communicate with the driver to create and assign volumes to a container.

A volume spec is needed to create a volume. A volume spec looks like:

```
// VolumeSpec has the properties needed to create a volume.
type VolumeSpec struct {
    // Ephemeral storage
    Ephemeral bool
    // Thin provisioned volume size in bytes
    Size uint64
    // Format disk with this FileSystem
    Format FileSystem
    // BlockSize for file system
    BlockSize int
    // HA Level specifies the number of nodes that are
    // allowed to fail, and yet data is available.
    // A value of 0 implies that data is not erasure coded,
    // a failure of a node will lead to data loss.
    HALevel int
    // This disk's CoS
    Cos VolumeCos
    // Perform dedupe on this disk
    Dedupe bool
    // SnapshotInterval in minutes, set to 0 to disable Snapshots
    SnapshotInterval int
    // Volume configuration labels
    ConfigLabels Labels
}
```

Various volume driver implementations can be found in the `drivers` directory.

Block Drivers

Block drivers operate at the block layer. They provide raw volumes formatted with a user specified filesystem. This volume is then mounted into the container at a path specified using the `docker run -v` option.

File Drivers

File drivers operate at the filesystem layer.