NAME

vdo - manage kernel VDO devices and related configuration information

SYNOPSIS

vdo { **activate** | **changeWritePolicy** | **create** | **deactivate** | **disableCompression** | **disableDeduplication** | **enableCompression** | **enableDeduplication** | **growLogical** | **growPhysical** | **list** | **modify** | **printConfigFile** | **remove** | **start** | **status** | **stop** } [options...]

DESCRIPTION

The commands available are:

activate

Activates one or more VDO volumes. Activated volumes can be started using the **start** command. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)
--**confFile=***file*
--**logfile=***file*
--**verbose**

changeWritePolicy

Modifies the write policy of one or all running VDO volumes. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)
--**writePolicy=**{ *sync* | *async* | *auto* } (required)
--**confFile=***file*
--**logfile=***file*
--**verbose**

create Creates a VDO volume and its associated index and makes it available. If --**activate=disabled** is specified the VDO volume is created but not made available. Will not overwrite an existing file system or formatted VDO volume unless --**force** is given. This command must be run with root privileges. Applicable options include:

--**name=***volume* (required)
--**device=***device* (required)
--**activate=**{ *enabled* | *disabled* }
--**blockMapCacheSize=***size*
--**blockMapPeriod=***period*
--**compression=**{ *enabled* | *disabled* }
--**deduplication=**{ *enabled* | *disabled* }
--**emulate512=**{ *enabled* | *disabled* }

```
--indexMem=size
--readCache={ enabled | disabled }
--readCacheSize=size
--sparseIndex={ enabled | disabled }
--vdoAckThreads=thread count
--vdoBioRotationInterval=I/O count
--vdoBioThreads=thread count
--vdoCpuThreads=thread count
--vdoHashZoneThreads=thread count
--vdoLogicalThreads=thread count
--vdoLogLevel=level
--vdoLogicalSize=size
--vdoPhysicalThreads=thread count
--vdoSlabSize=size
--writePolicy={ sync | async | auto }
--confFile=file
--logfile=file
--verbose
```

deactivate

Deactivates one or more VDO volumes. Deactivated volumes cannot be started by the **start** command. Deactivating a currently running volume does not stop it. Once stopped a deactivated VDO volume must be activated before it can be started again. This command must be run with root privileges. Applicable options include:

```
{ --all | --name=volume } (required)
--confFile=file
--logfile=file
--verbose
```

disableCompression

Disables compression on one or more VDO volumes. If the VDO volume is running, takes effect immediately. If the VDO volume is not running compression will be disabled the next time the VDO volume is started. This command must be run with root privileges. Applicable options include:

```
{ --all | --name=volume } (required)
--confFile=file
--logfile=file
--verbose
```

disableDeduplication

Disables deduplication on one or more VDO volumes. If the VDO volume is running, takes effect immediately. If the VDO volume is not running deduplication will be disabled the next time the VDO volume is started. This command must be run with root privileges. Applicable options include:

```
{ --all | --name=volume } (required)
--confFile=file
--logfile=file
--verbose
```

enableCompression

Enables compression on one or more VDO volumes. If the VDO volume is running, takes effect immediately. If the VDO volume is not running compression will be enabled the next time the VDO volume is started. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)

--**confFile=***file*

--**logfile=***file*

--**verbose**

enableDeduplication

Enables deduplication on one or more VDO volumes. If the VDO volume is running, takes effect immediately. If the VDO volume is not running deduplication will be enabled the next time the VDO volume is started. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)

--**confFile=***file*

--**logfile=***file*

--**verbose**

growLogical

Grows the logical size of a VDO volume. The volume must exist and must be running. This command must be run with root privileges. Applicable options include:

--**name=***volume* (required)

--**vdoLogicalSize=***size* (required)

--**confFile=***file*

--**logfile=***file*

--**verbose**

growPhysical

Grows the physical size of a VDO volume. The volume must exist and must be running. This command must be run with root privileges. Applicable options include:

--**name=***volume* (required)

--**confFile=***file*

--**verbose**

--**logfile=***file*

list Displays a list of started VDO volumes. If --**all** is specified it displays both started and non-started volumes. This command must be run with root privileges. Applicable options include:

--**all**

--**confFile=***file*

--**logfile=***file*

--**verbose**

modify Modifies configuration parameters of one or all VDO volumes. Changes take effect the next time the VDO device is started; already-running devices are not affected. Applicable options include:

{ --**all** | --**name=***volume* } (required)
--**blockMapCacheSize=***size*
--**blockMapPeriod=***period*
--**readCache=**{ *enabled* | *disabled* }
--**readCacheSize=***size*
--**vdoAckThreads=***thread count*
--**vdoBioThreads=***thread count*
--**vdoCpuThreads=***thread count*
--**vdoHashZoneThreads=***thread count*
--**vdoLogicalThreads=***thread count*
--**vdoPhysicalThreads=***thread count*
--**confFile=***file*
--**logfile=***file*
--**verbose**

printConfigFile

Prints the configuration file to stdout. This command does not require root privileges. Applicable options include:

--**confFile=***file*
--**logfile=***file*
--**verbose**

remove Removes one or more stopped VDO volumes and associated indexes. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)
--**force**
--**confFile=***file*
--**logfile=***file*
--**verbose**

start Starts one or more stopped, activated VDO volumes and associated services. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)
--**forceRebuild**
--**confFile=***file*
--**logfile=***file*
--**verbose**

status Reports VDO system and volume status in YAML format. This command does not require root privileges though information will be incomplete if run without. Applicable options include:

{ --**all** | --**name=***volume* }
--**confFile=***file*
--**logfile=***file*
--**verbose**

See below for the output provided.

stop Stops one or more running VDO volumes and associated services. This command must be run with root privileges. Applicable options include:

{ --**all** | --**name=***volume* } (required)

--**force**
--**confFile=**_file_
--**logfile=**_file_
--**verbose**

The **status** command returns the following information in YAML format, divided into keys as follows:

VDO Status
Information in this key covers the name of the host and date and time at which the status inquiry is being made. Parameters reported in this area include:

Node The host name of the system on which VDO is running.

Date The date and time at which the vdo status command is run.

Kernel Module
Information in this key covers the configured kernel.

Loaded Whether or not the kernel module is loaded (True or False).

Version Information
Information on the version of kvdo that is configured.

Configuration
Information in this key covers the location and status of the VDO configuration file.

File Location of the VDO configuration file.

Last modified
The last-modified date of the VDO configuration file.

VDOs Provides configuration information for all VDO volumes. Parameters reported for each VDO volume include:

Block size
The block size of the VDO volume, in bytes.

Emulate 512 byte
Indicates whether the volume is running in 512-byte emulation mode.

Deduplication
Whether deduplication is enabled for the volume.

Logical size
The logical size of the VDO volume.

Physical size
The size of a VDO volume's underlying physical storage.

Write policy

The configured value of the write policy (sync or async).

VDO Statistics

Output of the **vdostats** utility.

OPTIONS

The options supported by some or all of the commands listed above include:

--**activate=**{ *enabled* | *disabled* }

Indicates if the VDO volume should, in addition to being created, be activated and started. The default is **enabled.**

--**all**
-**a**

Indicates that the command should be applied to all configured VDO volumes. May not be used with --**name.**

--**blockMapCacheSize=***megabytes*

Specifies the amount of memory allocated for caching block map pages; the value must be a multiple of 4096. Using a value with a **B** (bytes), **K** (kilobytes), **M** (megabytes), **G** (gigabytes), **T** (terabytes), **P** (petabytes) or **E** (exabytes) suffix is optional. If no suffix is supplied, the value will be interpreted as **megabytes.** The value must be at least 128M and less than 16T. The cache must be at least 16MB per logical thread. Note that there is a memory overhead of 15%. The default is 128M.

--**blockMapPeriod=***period*

Tunes the quantity of block map updates that can accumulate before cache pages are flushed to disk. The value must at least 1 and less than or equal to 16380. A lower value means shorter recovery time but lower performance. The default value is 16380.

--**compression=**{ *enabled* | *disabled* }

Enables or disables compression when creating a VDO volume. The default is enabled. Compression may be disabled if necessary to maximize performance or to speed processing of data that is unlikely to compress.

--**confFile=***file*
-**f***file*

Specifies an alternate configuration file; the default is /etc/vdoconf.yml.

--**deduplication=**{ *enabled* | *disabled* }

Enables or disables deduplication when creating a VDO volume. The default is enabled. Deduplication may be disabled in instances where data is not expected to have good deduplication rates but compression is still desired.

--device=*absolute_path*

Specifies the absolute path of the device to use for VDO storage.

--emulate512={ *enabled* | *disabled* }

Specifies that the VDO volume is to emulate a 512 byte block device. The default is disabled.

--force

When creating a volume, ignores any existing file system or VDO signature already present in the storage device. When stopping or removing a VDO volume, first unmounts the file system stored on the device if mounted.

--forceRebuild

Forces an offline rebuild of a read-only VDO's metadata before starting so that it may be brought back online and made available. **This option may result in data loss or corruption.**

--indexMem=*gigabytes*

Specifies the amount of index memory in gigabytes; the default is currently 0.25 GB. The special decimal values 0.25, 0.5, 0.75 can be used, as can any integer value at least 1 and less than or equal to 1024. (The special deciml values are matched as exact strings; "0.5" works but "0.50" is not accepted.)

--help
-h

If specified with **vdo** only, displays documentation for the **vdo** utility. If specified with a command, displays documentation for that command.

--logfile=pathname

Specify the path of the file to which log messages are directed. If unspecified, log messages will go to syslog. Warning and error messages are always logged to syslog.

--name=*volume*
-n*volume*

Operates on the specified VDO volume. May not be used with **--all**.

--readCache={ *enabled* | *disabled* }

Enables or disables the read cache within the VDO device. The cache should be enabled if write workloads are expected to have high levels of deduplication, or for read intensive workloads of highly compressible data. The default is disabled.

--readCacheSize=*megabytes*

Specifies the extra VDO device read cache size in megabytes. This space is in addition to a system-defined minimum. Using a value with a **B** (bytes), **K** (kilobytes), **M** (megabytes), **G** (gigabytes), **T** (terabytes), **P** (petabytes) or **E** (exabytes) suffix is optional. The value must be at least 0 and less than 16T. 1.12 MB of memory will be used per MB of read cache specified, per bio thread. The default is 0.

--sparseIndex={ _enabled_ | _disabled_ }

Enables sparse indexing. The default is disabled.

--vdoAckThreads=_thread count_

Specifies the number of threads to use for acknowledging completion of requested VDO I/O operations. The value must be at least 0 and less than or equal to 100. The default is 1.

--vdoBioRotationInterval=_I/O count_

Specifies the number of I/O operations to enqueue for each bio-submission thread before directing work to the next. The value must be at least 1 and less than or equal to 1024. The default is 64.

--vdoBioThreads=_thread count_

Specifies the number of threads to use for submitting I/O operations to the storage device. The value must be at least 1 and less than or equal to 100. Each additional thread after the first will use an additional 18 MB of RAM, plus 1.12 MB of RAM per megabyte of configured read cache size. The default is 4.

--vdoCpuThreads=_thread count_

Specifies the number of threads to use for CPU-intensive work such as hashing or compression. The value must be at least 1 and less than or equal to 100. The default is 2.

--vdoHashZoneThreads=_thread count_

Specifies the number of threads across which to subdivide parts of the VDO processing based on the hash value computed from the block data. The value must be at least 1 and less than or equal to 100. vdoHashZonesThreads, vdoLogicalThreads and vdoPhysicalThreads must be either all zero or all non-zero. The default is 2.

--vdoLogicalThreads=_thread count_

Specifies the number of threads across which to subdivide parts of the VDO processing based on the hash value computed from the block data. The value must be at least 0 and less than or equal to 100. A logical thread count of 9 or more will require explicitly specifying a sufficiently large block map cache size, as well. vdoHashZonesThreads, vdoLogicalThreads and vdoPhysicalThreads must be either all zero or all non-zero. The default is 1.

--vdoLogicalSize=_megabytes_

Specifies the logical VDO volume size in megabytes. Using a value with a **S** (sectors), **B** (bytes), **K** (kilobytes), **M** (megabytes), **G** (gigabytes), **T** (terabytes), **P** (petabytes) or **E** (exabytes) suffix is optional. Used for over-provisioning volumes. The maximum size supported is 4P. The default is the size of the storage device.

--vdoLogLevel=*level*

Specifies the VDO driver log level: **critical**, **error**, **warning**, **notice**, **info**, or **debug**. Levels are case sensitive; the default is **info.**

--vdoPhysicalThreads=*thread count*

Specifies the number of threads across which to subdivide parts of the VDO processing based on physical block addresses. The value must be at least 0 and less than or equal to 16. Each additional thread after the first will use an additional 10 MB of RAM. vdoPhysicalThreads, vdoHashZonesThreads and vdoLogicalThreads must be either all zero or all non-zero. The default is 1.

--vdoSlabSize=*megabytes*

Specifies the size of the increment by which a VDO is grown. Using a smaller size constrains the total maximum physical size that can be accommodated. Must be a power of two between 128M and 32G. Using a value with a **S** (sectors), **B** (bytes), **K** (kilobytes), **M** (megabytes), **G** (gigabytes), **T** (terabytes), **P** (petabytes) or **E** (exabytes) suffix is optional. If no suffix is used, the value will be interpreted as megabytes. The default is 2G.

--verbose

Prints commands before executing them.

--writePolicy={ *sync* | *async* | *auto* }

Specifies the write policy:

sync Writes are acknowledged only after data is stably written. **This policy is not supported if the underlying storage is not also synchronous.**

async Writes are acknowledged after data has been *cached* for writing to stable storage. Data which has not been flushed is not guaranteed to persist in this mode.

auto VDO will check the storage device and determine whether it supports flushes. If it does, VDO will run in async mode, otherwise it will run in sync mode. This is the default.

FILES

/etc/vdoconf.yml

The default configuration file; used if the **--confFile** option is not provided.

EXAMPLES

Creation of a VDO device named **vdo0**, with a 10 terabyte thinly-provisioned logical address size:

```
# vdo create --name=vdo0 --device=/dev/sdb1 --vdoLogicalSize=10T
Creating VDO vdo0
Starting VDO vdo0
Starting compression on VDO vdo0
VDO instance 1 volume is ready at /dev/mapper/vdo0
#
```

Of course, as with any thinly-provisioned device, it may not hold 10 terabytes of user data even after deduplication and compression unless the underlying storage has sufficient space available for the resulting compressed, unique data blocks plus metadata overhead.

SEE ALSO

vdostats(8).