



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'podman-volume-create.1'

\$ man podman-volume-create.1

podman-volume-create(1) General Commands Manual podman-volume-create(1)

NAME

podman-volume-create - Create a new volume

SYNOPSIS

podman volume create [options] [name]

DESCRIPTION

Creates an empty volume and prepares it to be used by containers. The volume can be created with a specific name, if a name is not given a random name is generated. You can add metadata to the volume by using the --label flag and driver options can be set using the --opt flag.

OPTIONS

--driver, -d=driver

Specify the volume driver name (default local). There are two drivers supported by Podman itself: local and image. The local driver uses a directory on disk as the backend by default, but can also use the mount(8) command to mount a filesystem as the volume if --opt is specified. The image driver uses an image as the backing store of for the volume. An overlay filesystem will be created, which allows changes to

the volume to be committed as a new layer on top of the image. Using a value other than local or image, Podman will attempt to create the volume using a volume plugin with the given name. Such plugins must be defined in the volume_plugins section of the containers.conf(5) configuration file.

--help

Print usage statement

--ignore

Don't fail if the named volume already exists, instead just print the name. Note that the new options are not applied to the existing volume.

--label, -l=label

Set metadata for a volume (e.g., --label mykey=value).

--opt, -o=option

Set driver specific options. For the default driver, local, this allows a volume to be configured to mount a filesystem on the host. For the local driver the following options are supported: type, device, o, and [no]copy. The type option sets the type of the filesystem to be mounted, and is equivalent to the -t flag to mount(8). The device option sets the device to be mounted, and is equivalent to the device argument to mount(8). The copy option enables copying files from the container image path where the mount is created to the newly created volume on the first run. copy is the default.

The o option sets options for the mount, and is equivalent to the filesystem options (also -o) passed to mount(8) with the following exceptions:

? The o option supports uid and gid options to set the UID and GID of the created volume that are not normally supported by mount(8).

? The o option supports the size option to set the maximum size of the created volume, the inodes option to set the maximum number of inodes for the volume and noquota to completely disable quota support even for tracking of disk usage. Currently these flags are only supported on "xfs" file system mounted

with the `prjquota` flag described in the `xfs_quota(8)` man page.

? The `o` option supports using volume options other than the UID/GID options with the local driver and requires root privileges.

? The `o` options supports the `timeout` option which allows users to set a driver specific timeout in seconds before volume creation fails. For example, `--opts=o=timeout=10` sets a driver timeout of 10 seconds.

Note Do not confuse the `--opts,-o create` option with the `-o mount` option. For example, with `podman volume create`, use `-o=o=uid=1000` not `-o=uid=1000`.

For the image driver, the only supported option is `image`, which specifies the image the volume is based on. This option is mandatory when using the image driver.

When not using the local and image drivers, the given options are passed directly to the volume plugin. In this case, supported options are dictated by the plugin in question, not Podman.

EXAMPLES

```
$ podman volume create myvol
```

```
$ podman volume create
```

```
$ podman volume create --label foo=bar myvol
```

```
# podman volume create --opt device=tmpfs --opt type=tmpfs --opt o=nodev,noexec myvol
```

```
# podman volume create --opt device=tmpfs --opt type=tmpfs --opt o=uid=1000,gid=1000 testvol
```

```
# podman volume create --driver image --opt image=fedora:latest fedoraVol
```

QUOTAS

`podman volume create` uses XFS project quota controls for controlling the size and the number of inodes of builtin volumes. The directory used to store the volumes must be an XFS file system and be mounted with the `pquota` option.

Example `/etc/fstab` entry:

```
/dev/podman/podman-var /var xfs defaults,x-systemd.device-timeout=0,pquota 1 2
```

Podman generates project ids for each builtin volume, but these project ids need to be unique for the XFS file system. These project ids by de?

fault are generated randomly, with a potential for overlap with other quotas on the same file system.

The `xfstool` can be used to assign a project id to the storage driver directory, e.g.:

```
echo 100000:/var/lib/containers/storage/overlay >> /etc/projects
echo 200000:/var/lib/containers/storage/volumes >> /etc/projects
echo storage:100000 >> /etc/projid
echo volumes:200000 >> /etc/projid

xfstool -x -c 'project -s storage volumes' /<xfstool mount point>
```

In the example above we are configuring the overlay storage driver for newly created containers as well as volumes to use project ids with a start offset. All containers will be assigned larger project ids (e.g. `>= 100000`). All volume assigned project ids larger project ids starting with 200000. This prevents `xfstool` management conflicts with `containers/storage`.

SEE ALSO

[podman\(1\)](#), [containers.conf\(5\)](#), [podman-volume\(1\)](#), [mount\(8\)](#),
[xfstool\(8\)](#), [xfstool\(8\)](#), [projects\(5\)](#), [projid\(5\)](#)

HISTORY

January 2020, updated with information on volume plugins by Matthew Heon mheon@redhat.com?
November 2018, Originally compiled by Urvashi Mohnani umohnani@redhat.com?
umohnani@redhat.com?

[podman-volume-create\(1\)](#)