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

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'containers-storage.conf.5' command*

***$ man containers-storage.conf.5***

containers-storage.conf(5)(Container)Filecontainers-storage.conf(5)(Container)

Dan Walsh May 2017

NAME

storage.conf - Syntax of Container Storage configuration file

DESCRIPTION

The STORAGE configuration file specifies all of the available container storage options for tools using shared container storage, but in a TOML format that can be more easily modified and versioned.

FORMAT

The [TOML format][toml] is used as the encoding of the configuration file. Every option and subtable listed here is nested under a global "storage" table. No bare options are used. The format of TOML can be simplified to:

```
[table]
option = value
[table.subtable1]
option = value
[table.subtable2]
option = value
```

STORAGE TABLE

The storage table supports the following options:

driver=""

Copy On Write (COW) container storage driver. Valid drivers are

"overlay", "vfs", "devmapper", "aufs", "btrfs", and "zfs". Some drivers (for example, "zfs", "btrfs", and "aufs") may not work if your kernel lacks support for the filesystem. This field is required to guarantee proper operation. Valid rootless drivers are "btrfs", "overlay", and "vfs". Rootless users default to the driver defined in the system con? figuration when possible. When the system configuration uses an unsup? ported rootless driver, rootless users default to "overlay" if avail? able, otherwise "vfs".

graphroot=""

container storage graph dir (default: "/var/lib/containers/storage")

Default directory to store all writable content created by container storage programs. The rootless graphroot path supports environment variable substitutions (ie. $HOME/containers/storage). When changing the graphroot location on an SELINUX system, ensure the labeling matches the default locations labels with the following commands:

```
# semanage fcontext -a -e /var/lib/containers/storage /NEWSTORAGEPATH
# restorecon -R -v /NEWSTORAGEPATH
```

In rootless mode you would set

```
# semanage fcontext -a -e $HOME/.local/share/containers NEWSTORAGEPATH
$ restorecon -R -v /NEWSTORAGEPATH
```

rootless_storage_path="$HOME/.local/share/containers/storage"

Storage path for rootless users. By default the graphroot for root? less users is set to $XDG_DATA_HOME/containers/storage, if XDG_DATA_HOME is set. Otherwise $HOME/.local/share/containers/storage is used. This field can be used if administrators need to change the storage location for all users. The rootless storage path supports en? vironment variable substitutions (ie. $HOME/containers/storage)

A common use case for this field is to provide a local storage direc? tory when user home directories are NFS-mounted (podman does not sup? port container storage over NFS).

runroot=""

container storage run dir (default: "/run/containers/storage") De? fault directory to store all temporary writable content created by con?

tainer storage programs. The rootless runroot path supports environment variable substitutions (ie. $HOME/containers/storage)

driver_priority=[]

Priority list for the storage drivers that will be tested one after the other to pick the storage driver if it is not defined. The first storage driver in this list that can be used, will be picked as the new one and all subsequent ones will not be tried. If all drivers in this list are not viable, then all known drivers will be tried and the first working one will be picked. By default, the storage driver is set via the driver option. If it is not defined, then the best driver will be picked according to the current platform. This option allows you to override this internal priority list with a custom one to prefer cer? tain drivers. Setting this option only has an effect if the local storage has not been initialized yet and the driver name is not set.

STORAGE OPTIONS TABLE

The storage.options table supports the following options:

additionalimagestores=[]

Paths to additional container image stores. Usually these are read/only and stored on remote network shares.

pull_options = {enable_partial_images = "false", use_hard_links = "false", ostree_repos=""}

Allows specification of how storage is populated when pulling images. This option can speed the pulling process of images compressed with format zstd:chunked. Containers/storage looks for files within images that are being pulled from a container registry that were previously pulled to the host. It can copy or create a hard link to the existing file when it finds them, eliminating the need to pull them from the container registry. These options can deduplicate pulling of content, disk storage of content and can allow the kernel to use less memory when running containers.

containers/storage supports four keys

* enable_partial_images="true" | "false"

Tells containers/storage to look for files previously pulled in

storage

rather then always pulling them from the container registry.

* use_hard_links = "false" | "true"

Tells containers/storage to use hard links rather then create new files in

the image, if an identical file already existed in storage.

* ostree_repos = ""

Tells containers/storage where an ostree repository exists that might have

previously pulled content which can be used when attempting to avoid

pulling content from the container registry

remap-uids="" remap-gids=""

Remap-UIDs/GIDs is the mapping from UIDs/GIDs as they should appear inside of a container, to the UIDs/GIDs outside of the container, and the length of the range of UIDs/GIDs. Additional mapped sets can be listed and will be heeded by libraries, but there are limits to the number of mappings which the kernel will allow when you later attempt to run a container.

Example

remap-uids = 0:1668442479:65536

remap-gids = 0:1668442479:65536

These mappings tell the container engines to map UID 0 inside of the container to UID 1668442479 outside. UID 1 will be mapped to 1668442480. UID 2 will be mapped to 1668442481, etc, for the next 65533 UIDs in succession.

remap-user="" remap-group=""

Remap-User/Group is a user name which can be used to look up one or more UID/GID ranges in the /etc/subuid or /etc/subgid file. Mappings are set up starting with an in-container ID of 0 and then a host-level ID taken from the lowest range that matches the specified name, and us? ing the length of that range. Additional ranges are then assigned, us? ing the ranges which specify the lowest host-level IDs first, to the

lowest not-yet-mapped in-container ID, until all of the entries have been used for maps.

Example

remap-user = "containers"

remap-group = "containers"

root-auto-userns-user=""

Root-auto-userns-user is a user name which can be used to look up one or more UID/GID ranges in the /etc/subuid and /etc/subgid file. These ranges will be partitioned to containers configured to create automati? cally a user namespace. Containers configured to automatically create a user namespace can still overlap with containers having an explicit mapping set. This setting is ignored when running as rootless.

auto-userns-min-size=1024

Auto-userns-min-size is the minimum size for a user namespace created automatically.

auto-userns-max-size=65536

Auto-userns-max-size is the maximum size for a user namespace created automatically.

disable-volatile=true

If disable-volatile is set, then the "volatile" mount optimization is disabled for all the containers.

STORAGE OPTIONS FOR AUFS TABLE

The storage.options.aufs table supports the following options:

mountopt=""

Comma separated list of default options to be used to mount container images. Suggested value "nodev". Mount options are documented in the mount(8) man page.

STORAGE OPTIONS FOR BTRFS TABLE

The storage.options.btrfs table supports the following options:

min_space=""

Specifies the min space in a btrfs volume.

size=""

Maximum size of a container image. This flag can be used to set

quota on the size of container images. (format: [], where unit = b (bytes), k (kilobytes), m (megabytes), or g (gigabytes))

STORAGE OPTIONS FOR THINPOOL (devicemapper) TABLE

The storage.options.thinpool table supports the following options for the devicemapper driver:

autoextend_percent=""

Tells the thinpool driver the amount by which the thinpool needs to be grown. This is specified in terms of % of pool size. So a value of 20 means that when threshold is hit, pool will be grown by 20% of existing pool size. (default: 20%)

autoextend_threshold=""

Tells the driver the thinpool extension threshold in terms of percentage of pool size. For example, if threshold is 60, that means when pool is 60% full, threshold has been hit. (default: 80%)

basesize=""

Specifies the size to use when creating the base device, which limits the size of images and containers. (default: 10g)

blocksize=""

Specifies a custom blocksize to use for the thin pool. (default: 64k)

directlvm_device=""

Specifies a custom block storage device to use for the thin pool. Required for using graphdriver devicemapper.

directlvm_device_force=""

Tells driver to wipe device (directlvm_device) even if device already has a filesystem. (default: false)

fs="xfs"

Specifies the filesystem type to use for the base device. (default: xfs)

log_level=""

Sets the log level of devicemapper.

0: LogLevelSuppress 0 (default)

2: LogLevelFatal

3: LogLevelErr

4: LogLevelWarn

5: LogLevelNotice

6: LogLevelInfo

7: LogLevelDebug

metadata_size=""

metadata_size is used to set the pvcreate --metadatasize options when creating thin devices. (Default 128k)

min_free_space=""

Specifies the min free space percent in a thin pool required for new device creation to succeed. Valid values are from 0% - 99%. Value 0% disables. (default: 10%)

mkfsarg=""

Specifies extra mkfs arguments to be used when creating the base de? vice.

mountopt=""

Comma separated list of default options to be used to mount container images. Suggested value "nodev". Mount options are documented in the mount(8) man page.

size=""

Maximum size of a container image. This flag can be used to set quota on the size of container images. (format: [], where unit = b (bytes), k (kilobytes), m (megabytes), or g (gigabytes))

use_deferred_deletion=""

Marks thinpool device for deferred deletion. If the thinpool is in use when the driver attempts to delete it, the driver will attempt to delete device every 30 seconds until successful, or when it restarts. Deferred deletion permanently deletes the device and all data stored in the device will be lost. (default: true).

use_deferred_removal=""

Marks devicemapper block device for deferred removal. If the device is in use when its driver attempts to remove it, the driver tells the kernel to remove the device as soon as possible. Note this does not free up the disk space, use deferred deletion to fully remove the thin?

pool. (default: true).

xfs_nospace_max_retries=""

Specifies the maximum number of retries XFS should attempt to com? plete IO when ENOSPC (no space) error is returned by underlying storage device. (default: 0, which means to try continuously.)

STORAGE OPTIONS FOR OVERLAY TABLE

The storage.options.overlay table supports the following options:

ignore_chown_errors = "false"

ignore_chown_errors can be set to allow a non privileged user running with a single UID within a user namespace to run containers. The user can pull and use any image even those with multiple uids. Note multi? ple UIDs will be squashed down to the default uid in the container. These images will have no separation between the users in the con? tainer. (default: false)

inodes=""

Maximum inodes in a read/write layer. This flag can be used to set a quota on the inodes allocated for a read/write layer of a container.

force_mask = "0000|shared|private"

ForceMask specifies the permissions mask that is used for new files and directories. The values "shared" and "private" are accepted. (de? fault: ""). Octal permission masks are also accepted.

? ``: Not set All files/directories, get set with the permis? sions identified within the image.

? private: it is equivalent to 0700. All files/directories get set with 0700 permissions. The owner has rwx access to the files. No other users on the system can access the files. This setting could be used with networked based home directo? ries.

? shared: it is equivalent to 0755. The owner has rwx access to the files and everyone else can read, access and execute them. This setting is useful for sharing containers storage with other users. For instance, a storage owned by root could be shared to rootless users as an additional store. NOTE: All

files within the image are made readable and executable by any user on the system. Even /etc/shadow within your image is now readable by any user.

OCTAL: Users can experiment with other OCTAL Permissions.

Note: The force_mask Flag is an experimental feature, it could change in the future. When "force_mask" is set the original permission mask is stored in the "user.containers.override_stat" xattr and the "mount_program" option must be specified. Mount programs like "/usr/bin/fuse-overlayfs" present the extended attribute permissions to processes within containers rather than the "force_mask" permissions.

mount_program=""

Specifies the path to a custom program to use instead of using kernel defaults for mounting the file system. In rootless mode, without the CAP_SYS_ADMIN capability, many kernels prevent mounting of overlay file systems, requiring you to specify a mount_program. The mount_program option is also required on systems where the underlying storage is btrfs, aufs, zfs, overlay, or ecryptfs based file systems.

mount_program = "/usr/bin/fuse-overlayfs"

mountopt=""

Comma separated list of default options to be used to mount container images. Suggested value "nodev". Mount options are documented in the mount(8) man page.

skip_mount_home=""

Tell storage drivers to not create a PRIVATE bind mount on their home directory.

size=""

Maximum size of a read/write layer. This flag can be used to set quota on the size of a read/write layer of a container. (format: [], where unit = b (bytes), k (kilobytes), m (megabytes), or g (gigabytes))

STORAGE OPTIONS FOR VFS TABLE

The storage.options.vfs table supports the following options:

ignore_chown_errors = "false"

ignore_chown_errors can be set to allow a non privileged user running

with a single UID within a user namespace to run containers. The user can pull and use any image even those with multiple uids. Note multiple UIDs will be squashed down to the default uid in the container. These images will have no separation between the users in the container. (default: false)

STORAGE OPTIONS FOR ZFS TABLE

The storage.options.zfs table supports the following options:

fsname=""

File System name for the zfs driver

mountopt=""

Comma separated list of default options to be used to mount container images. Suggested value "nodev". Mount options are documented in the mount(8) man page.

size=""

Maximum size of a container image. This flag can be used to set quota on the size of container images. (format: [], where unit = b (bytes), k (kilobytes), m (megabytes), or g (gigabytes))

SELINUX LABELING

When running on an SELinux system, if you move the containers storage graphroot directory, you must make sure the labeling is correct.

Tell SELinux about the new containers storage by setting up an equivalence record. This tells SELinux to label content under the new path, as if it was stored under /var/lib/containers/storage.

```
semanage fcontext -a -e /var/lib/containers NEWSTORAGEPATH
restorecon -R -v NEWSTORAGEPATH
```

In rootless mode, you would set

```
semanage fcontext -a -e $HOME/.local/share/containers NEWSTORAGEPATH
restorecon -R -v NEWSTORAGEPATH
```

The semanage command above tells SELinux to setup the default labeling of NEWSTORAGEPATH to match /var/lib/containers. The restorecon command tells SELinux to apply the labels to the actual content.

Now all new content created in these directories will automatically be created with the correct label.

QUOTAS

Container storage implements XFS project quota controls for overlay storage containers and volumes. The directory used to store the con? tainers must be an XFS file system and be mounted with the pquota op? tion.

Example /etc/fstab entry:

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

Container storage generates project ids for each container and builtin volume, but these project ids need to be unique for the XFS file sys? tem.

The xfs_quota tool 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

xfs_quota -x -c 'project -s storage volumes' /<xfs mount point>

In the example above, the storage directory project id will be used as a "start offset" and all containers will be assigned larger project ids (e.g. >= 100000). Then the volumes directory project id will be used as a "start offset" and all volumes will be assigned larger project ids (e.g. >= 200000). This is a way to prevent xfs_quota management from conflicting with containers/storage.

FILES

Distributions often provide a /usr/share/containers/storage.conf file to define default storage configuration. Administrators can override this file by creating /etc/containers/storage.conf to specify their own configuration. Likewise rootless users can create a storage.conf file to override the system storage.conf files. Files should be stored in the $XDG_CONFIG_HOME/containers/storage.conf file. If $XDG_CONFIG_HOME is not set then the file $HOME/.config/containers/storage.conf is used.

Note: The storage.conf file overrides all other storage.conf files.

Container engines run by users with a storage.conf file in their home

directory do not use options in the system storage.conf files.

/etc/projects - XFS persistent project root definition /etc/projid -

XFS project name mapping file

SEE ALSO

semanage(8), restorecon(8), mount(8), fuse-overlayfs(1), xfs_quota(8),

projects(5), projid(5)

HISTORY

May 2017, Originally compiled by Dan Walsh dwalsh@redhat.com

?mailto:dwalsh@redhat.com? Format copied from crio.conf man page cre?

ated by Aleksa Sarai asarai@suse.de ?mailto:asarai@suse.de?

Configuration Storacontainers-storage.conf(5)(Container)