



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

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'buildah.1'***

**\$ man buildah.1**

buildah(1)                      General Commands Manual                      buildah(1)

#### NAME

Buildah - A command line tool that facilitates building OCI container images.

#### SYNOPSIS

buildah [OPTIONS] COMMAND [ARG...]

#### DESCRIPTION

The Buildah package provides a command line tool which can be used to:

- \* Create a working container, either from scratch or using an image as a starting point.
- \* Mount a working container's root filesystem for manipulation.
- \* Unmount a working container's root filesystem.
- \* Use the updated contents of a container's root filesystem as a filesystem layer to create a new image.
- \* Delete a working container or an image.
- \* Rename a local container.

#### OPTIONS

--log-level value

The log level to be used. Either "trace", "debug", "info", "warn", "error", "fatal", or "panic", defaulting to "warn".

--help, -h

Show help

--registries-conf path

Pathname of the configuration file which specifies which container registries should be consulted when completing image names which do not include a registry or domain portion.

It is not recommended that this option be used, as the default behavior of using the sys?

tem-wide configuration (/etc/containers/registries.conf) is most often preferred.

--registries-conf-dir path

Pathname of the directory which contains configuration snippets which specify registries which should be consulted when completing image names which do not include a registry or domain portion. It is not recommended that this option be used, as the default behavior of using the system-wide configuration (/etc/containers/registries.d) is most often preferred.

--root value

Storage root dir (default: "/var/lib/containers/storage" for UID 0, "\$HOME/.local/share/containers/storage" for other users) Default root dir is configured in /etc/containers/storage.conf

--runroot value

Storage state dir (default: "/run/containers/storage" for UID 0, "/run/user/\$UID" for other users) Default state dir is configured in /etc/containers/storage.conf

--short-name-alias-conf path

Pathname of the file which contains cached mappings between short image names and their corresponding fully-qualified names. It is used for mapping from names of images specified using short names like "hello-world" which don't include a registry component and a corresponding fully-specified name which includes a registry and any other components, such as "docker.io/library/hello-world". It is not recommended that this option be used, as the default behavior of using the system-wide cache (/var/cache/containers/short-name-aliases.conf) or per-user cache (\$HOME/.cache/containers/short-name-aliases.conf) to supplement system-wide defaults is most often preferred.

--storage-driver value

Storage driver. The default storage driver for UID 0 is configured in /etc/containers/storage.conf (\$HOME/.config/containers/storage.conf in rootless mode), and is vfs for other users. The STORAGE\_DRIVER environment variable overrides the default. The --storage-driver specified driver overrides all.

Examples: "overlay", "devicemapper", "vfs"

Overriding this option will cause the storage-opt settings in /etc/containers/storage.conf to be ignored. The user must specify additional options via the --storage-opt flag.

--storage-opt value

Storage driver option, Default storage driver options are configured in /etc/containers/storage.conf

ers/storage.conf (\$HOME/.config/containers/storage.conf in rootless mode). The STORAGE\_OPTS environment variable overrides the default. The --storage-opt specified options overrides all.

#### --users-uid-map mapping

Directly specifies a UID mapping which should be used to set ownership, at the filesystem level, on the working container's contents. Commands run when handling RUN instructions will default to being run in their own user namespaces, configured using the UID and GID maps.

Entries in this map take the form of one or more colon-separated triples of a starting in-container UID, a corresponding starting host-level UID, and the number of consecutive IDs which the map entry represents.

This option overrides the remap-uids setting in the options section of /etc/containers/storage.conf.

If this option is not specified, but a global --users-uid-map setting is supplied, settings from the global option will be used.

If none of --users-uid-map-user, --users-gid-map-group, or --users-uid-map are specified, but --users-gid-map is specified, the UID map will be set to use the same numeric values as the GID map.

NOTE: When this option is specified by a rootless user, the specified mappings are relative to the rootless usernamespace in the container, rather than being relative to the host as it would be when run rootful.

#### --users-gid-map mapping

Directly specifies a GID mapping which should be used to set ownership, at the filesystem level, on the working container's contents. Commands run when handling RUN instructions will default to being run in their own user namespaces, configured using the UID and GID maps.

Entries in this map take the form of one or more colon-separated triples of a starting in-container GID, a corresponding starting host-level GID, and the number of consecutive IDs which the map entry represents.

This option overrides the remap-gids setting in the options section of /etc/containers/storage.conf.

If this option is not specified, but a global --users-gid-map setting is supplied, settings from the global option will be used.

If none of --users-uid-map-user, --users-gid-map-group, or --users-gid-map are specified, but --users-uid-map is specified, the GID map will be set to use the same numeric values as the UID map.

NOTE: When this option is specified by a rootless user, the specified mappings are relative to the rootless usernamespace in the container, rather than being relative to the host as it would be when run rootful.

--version, -v

Print the version

### Environment Variables

Buildah can set up environment variables from the env entry in the [engine] table in the containers.conf(5). These variables can be overridden by passing environment variables before the buildah commands.

### COMMANDS

??

?Command ? Description ?

??

?buildah-add(1) ? Add the contents of a file, URL, ?

? ? or a directory to the container. ?

??

?buildah-build(1) ? Build an image using instructions ?

? ? tions from Dockerfiles. ?

??

?buildah-commit(1) ? Create an image from a working ?

? ? container. ?

??

?buildah-config(1) ? Update image configuration settings ?

? ? tings. ?

??

?buildah-containers(1) ? List the working containers and ?

? ? their base images. ?

??

?buildah-copy(1) ? Copies the contents of a file, ?

? ? URL, or directory into a container ?

? tainer's working directory. ?  
??  
?buildah-from(1) ? Creates a new working container, ?  
? either from scratch or using a ?  
? specified image as a starting ?  
? point. ?  
??  
?buildah-images(1) ? List images in local storage. ?  
??  
?buildah-info(1) ? Display Buildah system informa? ?  
? tion. ?  
??  
?buildah-inspect(1) ? Inspects the configuration of a ?  
? container or image ?  
??  
?buildah-login(1) ? Login to a container registry. ?  
??  
?buildah-logout(1) ? Logout of a container registry ?  
??  
?buildah-manifest(1) ? Create and manipulate manifest ?  
? lists and image indexes. ?  
??  
?buildah-mount(1) ? Mount the working container's ?  
? root filesystem. ?  
??  
?buildah-pull(1) ? Pull an image from the specified ?  
? location. ?  
??  
?buildah-push(1) ? Push an image from local storage ?  
? to elsewhere. ?  
??  
?buildah-rename(1) ? Rename a local container. ?  
??

?buildah-rm(1) ? Removes one or more working con? ?  
 ? ? tainers. ?  
 ???  
 ?buildah-rmi(1) ? Removes one or more images. ?  
 ???  
 ?buildah-run(1) ? Run a command inside of the con? ?  
 ? ? tainer. ?  
 ???  
 ?buildah-source(1) ? Create, push, pull and manage ?  
 ? ? source images and associated ?  
 ? ? source artifacts. ?  
 ???  
 ?buildah-tag(1) ? Add an additional name to a lo? ?  
 ? ? cal image. ?  
 ???  
 ?buildah-umount(1) ? Unmount a working container's ?  
 ? ? root file system. ?  
 ???  
 ?buildah-unshare(1) ? Launch a command in a user name? ?  
 ? ? space with modified ID mappings. ?  
 ???  
 ?buildah-version(1) ? Display the Buildah Version In? ?  
 ? ? formation ?  
 ???

Files

storage.conf (/etc/containers/storage.conf)

storage.conf is the storage configuration file for all tools using containers/storage

The storage configuration file specifies all of the available container storage options for tools using shared container storage.

mounts.conf (/usr/share/containers/mounts.conf and optionally /etc/containers/mounts.conf)

The mounts.conf files specify volume mount files or directories that are automatically mounted inside containers when executing the buildah run or buildah build-using-dockerfile

commands. Container processes can then use this content. The volume mount content does

not get committed to the final image.

Usually these directories are used for passing secrets or credentials required by the package software to access remote package repositories.

For example, a `mounts.conf` with the line `"/usr/share/rhel/secrets:/run/secrets"`, the content of `/usr/share/rhel/secrets` directory is mounted on `/run/secrets` inside the container.

This mountpoint allows Red Hat Enterprise Linux subscriptions from the host to be used within the container. It is also possible to omit the destination if it's equal to the source path. For example, specifying `/var/lib/secrets` will mount the directory into the same container destination path `/var/lib/secrets`.

Note this is not a volume mount. The content of the volumes is copied into container storage, not bind mounted directly from the host.

`registries.conf (/etc/containers/registries.conf)`

`registries.conf` is the configuration file which specifies which container registries should be consulted when completing image names which do not include a registry or domain portion.

`registries.d (/etc/containers/registries.d)`

Directory which contains configuration snippets which specify registries which should be consulted when completing image names which do not include a registry or domain portion.

## SEE ALSO

`containers.conf(5)`, `containers-mounts.conf(5)`, `newuidmap(1)`, `newgidmap(1)`, `containers-registries.conf(5)`, `containers-storage.conf(5)`

## HISTORY

December 2017, Originally compiled by Tom Sweeney [tsweeney@redhat.com](mailto:tsweeney@redhat.com)  
[?mailto:tsweeney@redhat.com?](mailto:tsweeney@redhat.com)

buildah

March 2017

buildah(1)