



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-add.1'

\$ man buildah-add.1

buildah-add(1) General Commands Manual buildah-add(1)

NAME

buildah-add - Add the contents of a file, URL, or a directory to a container.

SYNOPSIS

buildah add [options] container src [[src ...] dest]

DESCRIPTION

Adds the contents of a file, URL, or a directory to a container's working directory or a specified location in the container. If a local source file appears to be an archive, its contents are extracted and added instead of the archive file itself. If a local directory is specified as a source, its contents are copied to the destination.

OPTIONS

--add-history

Add an entry to the history which will note the digest of the added content. Defaults to false.

Note: You can also override the default value of --add-history by setting the BUILDDAH_HISTORY environment variable. export BUILDDAH_HISTORY=true

--chmod permissions

Sets the access permissions of the destination content. Accepts the numerical format.

--chown owner:group

Sets the user and group ownership of the destination content.

--contextdir directory

Build context directory. Specifying a context directory causes Buildah to chroot into that context directory. This means copying files pointed at by symbolic links outside of the

chroot will fail.

--from containerOrImage

Use the root directory of the specified working container or image as the root directory when resolving absolute source paths and the path of the context directory. If an image needs to be pulled, options recognized by buildah pull can be used.

--ignorefile file

Path to an alternative .containerignore (.dockerignore) file. Requires --contextdir be specified.

--quiet, -q

Refrain from printing a digest of the added content.

EXAMPLE

```
buildah add containerID '/myapp/app.conf' '/myapp/app.conf'
```

```
buildah add --chown myuser:mygroup containerID '/myapp/app.conf' '/myapp/app.conf'
```

```
buildah add --chmod 660 containerID '/myapp/app.conf' '/myapp/app.conf'
```

```
buildah add containerID '/home/myuser/myproject.go'
```

```
buildah add containerID '/home/myuser/myfiles.tar' '/tmp'
```

```
buildah add containerID '/tmp/workingdir' '/tmp/workingdir'
```

```
buildah add containerID 'https://github.com/containers/buildah/blob/main/README.md' '/tmp'
```

```
buildah add containerID 'passwd' 'certs.d' /etc
```

FILES

.containerignore/.dockerignore

If the .containerignore/.dockerignore file exists in the context directory, buildah add reads its contents. If both exist, then .containerignore is used.

When the \fB\fC--ignorefile\fR option is specified Buildah reads it and uses it to decide which content to exclude when copying content into the working container.

Users can specify a series of Unix shell glob patterns in an ignore file to identify files/directories to exclude.

Buildah supports a special wildcard string ** which matches any number of directories (including zero). For example, */.go will exclude all files that end with .go that are found in all directories.

Example .containerignore/.dockerignore file:

```
# here are files we want to exclude
```

```
*/*.c
```

`**/output*`

`src`

`*/*.c` Excludes files and directories whose names ends with `.c` in any top level subdirectory. For example, the source file `include/rootless.c`.

`**/output*` Excludes files and directories starting with `output` from any directory.

`src` Excludes files named `src` and the directory `src` as well as any content in it.

Lines starting with `!` (exclamation mark) can be used to make exceptions to exclusions. The following is an example `.containerignore` file that uses this mechanism:

`*.doc`

`!Help.doc`

Exclude all `doc` files except `Help.doc` when copying content into the container.

This functionality is compatible with the handling of `.dockerignore` files described here:

<https://docs.docker.com/engine/reference/builder/#dockerignore-file>

SEE ALSO

`buildah(1)`

`buildah`

April 2021

`buildah-add(1)`