



## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'containerignore.5' command***

### ***\$ man containerignore.5***

.containerignore(28)      Container User Manuals      .containerignore(28)

#### **NAME**

.containerignore(.dockerignore) - files to ignore buildah or podman build context directory

#### **INTRODUCTION**

Before container engines build an image, they look for a file named .containerignore or .dockerignore in the root context directory. If one of these file exists, the CLI modifies the context to exclude files and directories that match patterns specified in the file. This avoids adding them to images using the ADD or COPY instruction.

The CLI interprets the .containerignore or .dockerignore file as a new? line-separated list of patterns similar to the file globs of Unix shells. For the purposes of matching, the root of the context is considered to be both the working and the root directory. For example, the patterns /foo/bar and foo/bar both exclude a file or directory named bar in the foo subdirectory of PATH or in the root of the git repository located at URL. Neither excludes anything else.

If a line in .containerignore or .dockerignore file starts with # in column 1, then this line is considered as a comment and is ignored before interpreted by the CLI.

#### **EXAMPLES**

Here is an example .containerignore file:

```
# comment
```

`*/temp*`

`*/*/temp*`

`temp?`

This file causes the following build behavior: Rule      Behavior

`# comment`      Ignored.

`*/temp*`      Exclude files and directories whose names start with temp in any immediate subdirectory of the root.

For example, the plain file `/somedir/temporary.txt` is excluded, as is the directory `/somedir/temp`.

`*/*/temp*`      Exclude files and directories starting with temp from any subdirectory that is two levels below the root. For example, `/somedir/subdir/temporary.txt` is excluded.

`temp?`      Exclude files and directories in the root directory whose names are a one-character extension of temp.

For example, `/tempa` and `/tempb` are excluded.

Matching is done using Go's `filepath.Match` rules. A preprocessing step removes leading and trailing whitespace and eliminates `.` and `..` elements using Go's `filepath.Clean`. Lines that are blank after preprocessing are ignored.

Beyond Go's `filepath.Match` rules, Docker also supports a special wildcard string `**` that matches any number of directories (including zero).

For example, `*/.go` will exclude all files that end with `.go` that are found in all directories, including the root of the build context.

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

`*.md`

`!README.md`

All markdown files except `README.md` are excluded from the context.

The placement of `!` exception rules influences the behavior: the last line of the `.containerignore` that matches a particular file determines whether it is included or excluded. Consider the following example:

`*.md`

`!README*.md`

`README-secret.md`

No markdown files are included in the context except `README` files other than `README-secret.md`.

Now consider this example:

```
*.md
```

```
README-secret.md
```

```
!README*.md
```

All of the README files are included. The middle line has no effect be?

cause !README\*.md matches README-secret.md and comes last.

You can even use the .containerignore file to exclude the Containerfile

or Dockerfile and .containerignore files. These files are still sent

to the daemon because it needs them to do its job. But the ADD and COPY

instructions do not copy them to the image.

Finally, you may want to specify which files to include in the context,

rather than which to exclude. To achieve this, specify \* as the first

pattern, followed by one or more ! exception patterns.

## SEE ALSO

buildah-build(1), podman-build(1), docker-build(1)

## HISTORY

\*Sep 2021, Compiled by Dan Walsh (dwalsh at redhat dot com) based on  
docker.com .dockerignore documentation.

Sep 2021 .containerignore(28)