



Rocky Enterprise Linux 9.2 Manual Pages on command 'buildah-manifest.1'

\$ man buildah-manifest.1

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

NAME

buildah-manifest - Create and manipulate manifest lists and image indexes.

SYNOPSIS

buildah manifest COMMAND [OPTIONS] [ARG...]

DESCRIPTION

The buildah manifest command provides subcommands which can be used to:

- * Create a working Docker manifest list or OCI image index.
- * Add an entry to a manifest list or image index for a specified image.
- * Add or update information about an entry in a manifest list or image index.
- * Delete a working container or an image.
- * Push a manifest list or image index to a registry or other location.

SUBCOMMANDS

??

?Command	? Man Page	? Description	?
----------	------------	---------------	---

??

?add	? buildah-manifest-add(1)	? Add an image to a ?	
------	---------------------------	-----------------------	--

? ? ? manifest list or ?

? ? ? image index. ?

??

?annotate ? buildah-manifest-annotate(1) ? Add or update in? ?

? ? ? formation about an ?

? ? ? image in a manifest ?

? ? ? list or image in? ?

? ? ? dex. ?

??

?create ? buildah-manifest-create(1) ? Create a manifest ?

? ? ? list or image in? ?

? ? ? dex. ?

??

?exists ? buildah-manifest-exists(1) ? Check if a manifest ?

? ? ? list exists in lo? ?

? ? ? cal storage. ?

??

?inspect ? buildah-manifest-inspect(1) ? Display the con? ?

? ? ? tents of a manifest ?

? ? ? list or image in? ?

? ? ? dex. ?

??

?push ? buildah-manifest-push(1) ? Push a manifest ?

? ? ? list or image index ?

? ? ? to a registry or ?

? ? ? other location. ?

??

?remove ? buildah-manifest-remove(1) ? Remove an image ?

? ? ? from a manifest ?

? ? ? list or image in? ?

? ? ? dex. ?

??

?rm ? buildah-manifest-rm(1) ? Remove manifest ?

```
?      ?      ? list from local ?
?      ?      ? storage.      ?
????????????????????????????????????????????????????????????
```

EXAMPLES

Building a multi-arch manifest list from a Containerfile

Assuming the Containerfile uses RUN instructions, the host needs a way to execute non-native binaries. Configuring this is beyond the scope of this example. Building a multi-arch manifest list shazam in parallel across 4-threads can be done like this:

```
$ platarch=linux/amd64,linux/ppc64le,linux/arm64,linux/s390x
$ buildah build --jobs=4 --platform=$platarch --manifest shazam .
```

Note: The --jobs argument is optional, and the -t or --tag option should not be used.

Assembling a multi-arch manifest from separately built images

Assuming example.com/example/shazam:\$arch images are built separately on other hosts and pushed to the example.com registry. They may be combined into a manifest list, and pushed using a simple loop:

```
$ REPO=example.com/example/shazam
$ buildah manifest create $REPO:latest
$ for IMGTAG in amd64 s390x ppc64le arm64; do
    buildah manifest add $REPO:latest docker://$REPO:IMGTAG;
done
$ buildah manifest push --all $REPO:latest
```

Note: The add instruction argument order is <manifest> then <image>. Also, the --all push option is required to ensure all contents are pushed, not just the native platform/arch.

Removing and tagging a manifest list before pushing

Special care is needed when removing and pushing manifest lists, as opposed to the contents. You almost always want to use the manifest rm and manifest push --all subcommands. For example, a rename and push could be performed like this:

```
$ buildah tag localhost/shazam example.com/example/shazam
$ buildah manifest rm localhost/shazam
```

```
$ buildah manifest push --all example.com/example/shazam
```

SEE ALSO

buildah(1), buildah-manifest-create(1), buildah-manifest-add(1), buildah-manifest-remove(1), buildah-manifest-annotate(1), buildah-manifest-inspect(1), buildah-manifest-push(1), buildah-manifest-rm(1)

buildah September 2019 buildah-manifest(1)