



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 'podman-manifest.1'

\$ man podman-manifest.1

podman-manifest(1)() podman-manifest(1)()

NAME

podman-manifest - Create and manipulate manifest lists and image indexes

SYNOPSIS

podman manifest subcommand

DESCRIPTION

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

- * Create a working Docker manifest list or OCI image index.

SUBCOMMANDS

??

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

??

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

?	?	? fest list or image in?	?
---	---	--------------------------	---

?	?	? dex.	?
---	---	--------	---

??

?annotate	? podman-manifest-annotate(1)	? Add or update informa?	?
-----------	-------------------------------	--------------------------	---

?	?	? tion about an entry in a ?	
---	---	------------------------------	--

?	?	? manifest list or image ?	
---	---	----------------------------	--

?	?	? index.	?
---	---	----------	---

??

?create	? podman-manifest-create(1)	? Create a manifest list ?	
---------	-----------------------------	----------------------------	--

?	?	? or image index.	?
---	---	-------------------	---

??

?exists ? podman-manifest-exists(1) ? Check if the given mani? ?

? ? ? fest list exists in lo? ?

? ? ? cal storage ?

??

?inspect ? podman-manifest-inspect(1) ? Display a manifest list ?

? ? ? or image index. ?

??

?push ? podman-manifest-push(1) ? Push a manifest list or ?

? ? ? image index to a reg? ?

? ? ? istry. ?

??

?remove ? podman-manifest-remove(1) ? Remove an image from a ?

? ? ? manifest list or image ?

? ? ? index. ?

??

?rm ? podman-manifest-rme(1) ? Remove manifest list or ?

? ? ? image index 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
```

```
$ podman 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
```

```
$ podman manifest create $REPO:latest
$ for IMGTAG in amd64 s390x ppc64le arm64; do
    podman manifest add $REPO:latest docker://$REPO:IMGTAG;
done
$ podman 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:

```
$ podman tag localhost/shazam example.com/example/shazam
$ podman manifest rm localhost/shazam
$ podman manifest push --all example.com/example/shazam
```

SEE ALSO

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