Rocky Enterprise Linux 9.2 Manual Pages on command 'docker-image-prune.1'

$ man docker-image-prune.1

podman-image-prune(1)()   podman-image-prune(1)()

NAME

   podman-image-prune - Remove all unused images from the local store

SYNOPSIS

   podman image prune [options]

DESCRIPTION

   podman image prune removes all dangling images from local storage. With the all option,

   you can delete all unused images (i.e., images not in use by any container).

   The image prune command does not prune cache images that only use layers that are neces?

   sary for other images.

OPTIONS

  --all, -a

   Remove dangling images and images that have no associated containers.

  --filter=filters

   Provide filter values.

   The --filter flag format is of ?key=value?. If there is more than one filter, then pass

   multiple flags (e.g., --filter "foo=bar" --filter "bif=baz")

   Supported filters:

       ? until (timestamp) - only remove containers and images created before given time?

         stamp

       ? label (label=key, label=key=value, label!=key, or label!=key=value) - only remove

         containers and images, with (or without, in case label!=... is used) the speci?

         fied labels.

The until filter can be Unix timestamps, date formatted timestamps, or Go duration strings

(e.g. 10m, 1h30m) computed relative to the machine?s time.

The label filter accepts two formats. One is the label=... (label=key or label=key=value),

which removes images with the specified labels. The other format is the label!=... (la?

bel!=key or label!=key=value), which removes images without the specified labels.

--force, -f

Do not provide an interactive prompt for container removal.

--help, -h

Print usage statement

EXAMPLES

Remove all dangling images from local storage

$ sudo podman image prune

WARNING! This will remove all dangling images.

Are you sure you want to continue? [y/N] y

f3e20dc537fb04cb51672a5cb6fdf2292e61d411315549391a0d1f64e4e3097e

324a7a3b2e0135f4226ffdd473e4099fd9e477a74230cdc35de69e84c0f9d907

Remove all unused images from local storage without confirming

$ sudo podman image prune -a -f

f3e20dc537fb04cb51672a5cb6fdf2292e61d411315549391a0d1f64e4e3097e

324a7a3b2e0135f4226ffdd473e4099fd9e477a74230cdc35de69e84c0f9d907

6125002719feb1ddf3030acab1df6156da7ce0e78e571e9b6e9c250424d6220c

91e732da5657264c6f4641b8d0c4001c218ae6c1adb9dcef33ad00cafd37d8b6

e4e5109420323221f170627c138817770fb64832da7d8fe2babd863148287fca

77a57fa8285e9656dbb7b23d9efa837a106957409ddd702f995605af27a45ebe

Remove all unused images from local storage since given time/hours.

$ sudo podman image prune -a --filter until=2019-11-14T06:15:42.937792374Z

WARNING! This will remove all dangling images.

Are you sure you want to continue? [y/N] y

e813d2135f17fadeffeea8159a34cfdd4c30b98d8111364b913a91fd930643e9

5e6572320437022e2746467ddf5b3561bf06e099e8e6361df27e0b2a7ed0b17b

58fda2abf5042b35dfe04e5f8ee458a3cc26375bf309efb42c078b551a2055c7

6d2bd30fe924d3414b64bd3920760617e6ced872364bc3bc6959a623252da002

33d1c829be64a1e1d379caf4feec1f05a892c3ef7aa82c0be53d3c08a96c59c5

```
f9f0a8a58c9e02a2b3250b88cc5c95b1e10245ca2c4161d19376580aaa90f55c
1ef14d5ede80db78978b25ad677fd3e897a578c3af614e1fda608d40c8809707
45e1482040e441a521953a6da2eca9bafc769e15667a07c23720d6e0cafc3ab2
$ sudo podman image prune -f --filter until=10h
f3e20dc537fb04cb51672a5cb6fdf2292e61d411315549391a0d1f64e4e3097e
324a7a3b2e0135f4226ffdd473e4099fd9e477a74230cdc35de69e84c0f9d907
```

Remove all unused images from local storage with label version 1.0

```
$ sudo podman image prune -a -f --filter label=version=1.0
e813d2135f17fadeffeea8159a34cfdd4c30b98d8111364b913a91fd930643e9
5e6572320437022e2746467ddf5b3561bf06e099e8e6361df27e0b2a7ed0b17b
58fda2abf5042b35dfe04e5f8ee458a3cc26375bf309efb42c078b551a2055c7
6d2bd30fe924d3414b64bd3920760617e6ced872364bc3bc6959a623252da002
33d1c829be64a1e1d379caf4feec1f05a892c3ef7aa82c0be53d3c08a96c59c5
f9f0a8a58c9e02a2b3250b88cc5c95b1e10245ca2c4161d19376580aaa90f55c
1ef14d5ede80db78978b25ad677fd3e897a578c3af614e1fda608d40c8809707
45e1482040e441a521953a6da2eca9bafc769e15667a07c23720d6e0cafc3ab2
```

SEE ALSO

podman(1), podman-images

HISTORY

December 2018, Originally compiled by Brent Baude (bbaude at redhat dot com) December 2020, converted filter information from docs.docker.com documentation by Dan Walsh (dwalsh at redhat dot com)

<div align="center">podman-image-prune(1)()</div>