Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'podman-image-pull.1' command

$ man podman-image-pull.1

podman-pull(1) General Commands Manual podman-pull(1)

NAME

podman-pull - Pull an image from a registry

SYNOPSIS

podman pull [options] source [source...]

podman image pull [options] source [source...]

podman pull [options] [transport]name[:tag|@digest]

podman image pull [options] [transport]name[:tag|@digest]

DESCRIPTION

podman pull copies an image from a registry onto the local machine. The command can pull one or more images. If the image reference in the command line argument does not contain a registry, it is referred to as ashort-name reference. If the image is a 'short-name' reference, Podman will prompt the user for the specific container registry to pull the image from, if an alias for the short-name has not been specified in the short-name-aliases.conf. If an image tag is not specified, podman pull defaults to the image with the latest tag (if it exists) and pulls it. After the image is pulled, podman will print the full image ID. podman pull can also pull images using a digest podman pull image@di? gest and can also be used to pull images from archives and local stor? age using different transports. IMPORTANT: Images are stored in local image storage.

SOURCE

SOURCE is the location from which the container image is pulled from. It supports all transports from containers-transports(5). If no trans? port is specified, the input is subject to short-name resolution and the docker (i.e., container registry) transport is used. For remote clients, including Mac and Windows (excluding WSL2) machines, docker is the only supported transport.

```
# Pull from a container registry
$ podman pull quay.io/username/myimage
# Pull from a container registry with short-name resolution
$ podman pull fedora
# Pull from a container registry via the docker transport
$ podman pull docker://quay.io/username/myimage
# Pull from a local directory
$ podman pull dir:/tmp/myimage
# Pull from a tarball in the docker-archive format
$ podman pull docker-archive:/tmp/myimage
# Pull from a local docker daemon
$ sudo podman pull docker-daemon:docker.io/library/myimage:33
# Pull from a tarball in the OCI-archive format
$ podman pull oci-archive:/tmp/myimage
```

OPTIONS

--all-tags, -a

All tagged images in the repository will be pulled.

IMPORTANT: When using the all-tags flag, Podman will not iterate over the search registries in the containers-registries.conf(5) but will al? ways use docker.io for unqualified image names.

--arch=ARCH

Override the architecture, defaults to hosts, of the image to be pulled. For example, arm. Unless overridden, subsequent lookups of the same image in the local storage will match this architecture, regard? less of the host.

--authfile=path

Path of the authentication file. Default is ${XDG_RUNTIME_DIR}/contain?

ers/auth.json, which is set using podman login. If the authorization state is not found there, $HOME/.docker/config.json is checked, which is set using docker login.

Note: There is also the option to override the default path of the au? thentication file by setting the REGISTRY_AUTH_FILE environment vari? able. This can be done with export REGISTRY_AUTH_FILE=path.

--cert-dir=path

Use certificates at path (*.crt, *.cert, *.key) to connect to the reg? istry. (Default: /etc/containers/certs.d) Please refer to containers-certs.d(5) for details. (This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2) machines)

--creds=[username[:password]]

The [username[:password]] to use to authenticate with the registry, if required. If one or both values are not supplied, a command line prompt will appear and the value can be entered. The password is en? tered without echo.

--decryption-key=key[:passphrase]

The [key[:passphrase]] to be used for decryption of images. Key can point to keys and/or certificates. Decryption will be tried with all keys. If the key is protected by a passphrase, it is required to be passed in the argument and omitted otherwise.

--disable-content-trust

This is a Docker-specific option to disable image verification to a container registry and is not supported by Podman. This option is a NOOP and provided solely for scripting compatibility.

--help, -h

Print the usage statement.

--os=OS

Override the OS, defaults to hosts, of the image to be pulled. For ex? ample, windows. Unless overridden, subsequent lookups of the same im? age in the local storage will match this OS, regardless of the host.

--platform=OS/ARCH

Specify the platform for selecting the image. (Conflicts with --arch

and --os) The --platform option can be used to override the current ar? chitecture and operating system. Unless overridden, subsequent lookups of the same image in the local storage will match this platform, re? gardless of the host.

--quiet, -q

Suppress output information when pulling images

--tls-verify

Require HTTPS and verify certificates when contacting registries (de? fault: true). If explicitly set to true, TLS verification will be used. If set to false, TLS verification will not be used. If not specified, TLS verification will be used unless the target registry is listed as an insecure registry in containers-registries.conf(5)

--variant=VARIANT

Use VARIANT instead of the default architecture variant of the con? tainer image. Some images can use multiple variants of the arm archi? tectures, such as arm/v5 and arm/v7.

FILES

short-name-aliases.conf (/var/cache/containers/short-name-aliases.conf, $HOME/.cache/containers/short-name-aliases.conf)

When users specify images that do not include the container registry where the image is stored, this is called a short name. The use of un? qualified-search registries entails an ambiguity as it is unclear from which registry a given image, referenced by a short name, may be pulled from.

Using short names is subject to the risk of hitting squatted registry namespaces. If the unqualified-search registries are set to ["public-registry.com", "my-private-registry.com"] an attacker may take over a namespace of public-registry.com such that an image may be pulled from public-registry.com instead of the intended source my-private-reg? istry.com.

While it is highly recommended to always use fully-qualified image ref? erences, existing deployments using short names may not be easily changed. To circumvent the aforementioned ambiguity, so called short-

name aliases can be configured that point to a fully-qualified image reference. Distributions often ship a default shortnames.conf expansion file in /etc/containers/registries.conf.d/ directory. Administrators can use this directory to add their own local short-name expansion files.

When pulling an image, if the user does not specify the complete registry, container engines attempt to expand the short-name into a full name. If the command is executed with a tty, the user will be prompted to select a registry from the default list unqualified registries defined in registries.conf. The user's selection is then stored in a cache file to be used in all future short-name expansions. Rootful short-names are stored in /var/cache/containers/short-name-aliases.conf. Rootless short-names are stored in the $HOME/.cache/containers/short-name-aliases.conf file.

For more information on short-names, see containers-registries.conf(5)

registries.conf (/etc/containers/registries.conf)

registries.conf is the configuration file which specifies which container registries should be consulted when completing image names which do not include a registry or domain portion.

NOTE: Use the environment variable TMPDIR to change the temporary storage location of downloaded container images. Podman defaults to use /var/tmp.

EXAMPLES

Pull a single image with short name resolution.

```
$ podman pull alpine:latest
Resolved "alpine" as an alias (/etc/containers/registries.conf.d/000-shortnames.conf)
Trying to pull docker.io/library/alpine:latest...
Getting image source signatures
Copying blob 5843afab3874 done
Copying config d4ff818577 done
Writing manifest to image destination
Storing signatures
d4ff818577bc193b309b355b02ebc9220427090057b54a59e73b79bdfe139b83
```

Pull multiple images with/without short name resolution.

```
podman pull busybox:musl alpine quay.io/libpod/cirros
Trying to pull docker.io/library/busybox:musl...
Getting image source signatures
Copying blob 0c52b060233b [--------------------------------------] 0.0b / 0.0b
Copying config 9ad2c435a8 done
Writing manifest to image destination
Storing signatures
9ad2c435a887e3f723654e09b48563de44aa3c7950246b2e9305ec85dd3422db
Trying to pull docker.io/library/alpine:latest...
Getting image source signatures
Copying blob 5843afab3874 [--------------------------------------] 0.0b / 0.0b
Copying config d4ff818577 done
Writing manifest to image destination
Storing signatures
d4ff818577bc193b309b355b02ebc9220427090057b54a59e73b79bdfe139b83
Trying to pull quay.io/libpod/cirros:latest...
Getting image source signatures
Copying blob 8da581cc9286 done
Copying blob 856628d95d17 done
Copying blob f513001ba4ab done
Copying config 3c82e4d066 done
Writing manifest to image destination
Storing signatures
3c82e4d066cf6f9e50efaead6e3ff7fddddf5527826afd68e5a969579fc4db4a
```

Pull an image using its digest.

```
$ podman pull alpine@sha256:d7342993700f8cd7aba8496c2d0e57be0666e80b4c441925fc6f9361fa81d10e
Trying to pull docker.io/library/alpine@sha256:d7342993700f8cd7aba8496c2d0e57be0666e80b4c441925fc6f9361fa81d10e...
Getting image source signatures
Copying blob 188c0c94c7c5 done
Copying config d6e46aa247 done
Writing manifest to image destination
```

Storing signatures

d6e46aa2470df1d32034c6707c8041158b652f38d2a9ae3d7ad7e7532d22ebe0

Pull an image by specifying an authentication file.

$ podman pull --authfile temp-auths/myauths.json docker://docker.io/umohnani/finaltest

Trying to pull docker.io/umohnani/finaltest:latest...Getting image source signatures

Copying blob sha256:6d987f6f42797d81a318c40d442369ba3dc124883a0964d40b0c8f4f7561d913

1.90 MB / 1.90 MB [===] 0s

Copying config sha256:ad4686094d8f0186ec8249fc4917b71faa2c1030d7b5a025c29f26e19d95c156

1.41 KB / 1.41 KB [===] 0s

Writing manifest to image destination

Storing signatures

03290064078cb797f3e0a530e78c20c13dd22a3dd3adf84a5da2127b48df0438

Pull an image by authenticating to a registry.

$ podman pull --creds testuser:testpassword docker.io/umohnani/finaltest

Trying to pull docker.io/umohnani/finaltest:latest...Getting image source signatures

Copying blob sha256:6d987f6f42797d81a318c40d442369ba3dc124883a0964d40b0c8f4f7561d913

1.90 MB / 1.90 MB [===] 0s

Copying config sha256:ad4686094d8f0186ec8249fc4917b71faa2c1030d7b5a025c29f26e19d95c156

1.41 KB / 1.41 KB [===] 0s

Writing manifest to image destination

Storing signatures

03290064078cb797f3e0a530e78c20c13dd22a3dd3adf84a5da2127b48df0438

Pull an image using tls verification.

$ podman pull --tls-verify=false --cert-dir image/certs docker.io/umohnani/finaltest

Trying to pull docker.io/umohnani/finaltest:latest...Getting image source signatures

Copying blob sha256:6d987f6f42797d81a318c40d442369ba3dc124883a0964d40b0c8f4f7561d913

1.90 MB / 1.90 MB [===] 0s

Copying config sha256:ad4686094d8f0186ec8249fc4917b71faa2c1030d7b5a025c29f26e19d95c156

1.41 KB / 1.41 KB [===] 0s

Writing manifest to image destination

Storing signatures

03290064078cb797f3e0a530e78c20c13dd22a3dd3adf84a5da2127b48df0438

Pull an image by overriding the host architecture.

$ podman pull --arch=arm arm32v7/debian:stretch

Trying to pull docker.io/arm32v7/debian:stretch...

Getting image source signatures

Copying blob b531ae4a3925 done

Copying config 3cba58dad5 done

Writing manifest to image destination

Storing signatures

3cba58dad5d9b35e755b48b634acb3fdd185ab1c996ac11510cc72c17780e13c

SEE ALSO

podman(1), podman-push(1), podman-login(1), containers-certs.d(5), con?

tainers-registries.conf(5), containers-transports(5)

HISTORY

July 2017, Originally compiled by Urvashi Mohnani umohnani@redhat.com

?mailto:umohnani@redhat.com?

podman-pull(1)