Rocky Enterprise Linux 9.2 Manual Pages on command 'containers-transports.5'

$ man containers-transports.5

CONTAINERS-TRANSPORTS(5)   Man   CONTAINERS-TRANSPORTS(5)

Valentin Rothberg April 2019

NAME

containers-transports - description of supported transports for copying and storing con?
tainer images

DESCRIPTION

Tools which use the containers/image library, including skopeo(1), buildah(1), podman(1),
all share a common syntax for referring to container images in various locations. The
general form of the syntax is transport:details, where details are dependent on the speci?
fied transport, which are documented below.

containers-storage:[[storage-specifier]]{image-id|docker-reference[@image-id]}

An image located in a local containers storage. The format of docker-reference is de?
scribed in detail in the docker transport.

The storage-specifier allows for referencing storage locations on the file system and has
the format [[driver@]root[+run-root][:options]] where the optional driver refers to the
storage driver (e.g., overlay or btrfs) and where root is an absolute path to the stor?
age's root directory. The optional run-root can be used to specify the run directory of
the storage where all temporary writable content is stored. The optional options are a
comma-separated list of driver-specific options.   Please refer to containers-stor?
age.conf(5) for further information on the drivers and supported options.

dir:path

An existing local directory path storing the manifest, layer tarballs and signatures as
individual files.   This is a non-standardized format, primarily useful for debugging or

noninvasive container inspection.

docker://docker-reference

An image in a registry implementing the "Docker Registry HTTP API V2". By default, uses the authorization state in $XDG_RUNTIME_DIR/containers/auth.json, which is set using pod?man-login(1). If the authorization state is not found there, $HOME/.docker/config.json is checked, which is set using docker-login(1). The containers-registries.conf(5) further allows for configuring various settings of a registry.

Note that a docker-reference has the following format: name[:tag|@digest]. While the docker transport does not support both a tag and a digest at the same time some formats like containers-storage do. Digests can also be used in an image destination as long as the manifest matches the provided digest. The digest of images can be explored with skopeo-inspect(1). If name does not contain a slash, it is treated as docker.io/li?brary/name. Otherwise, the component before the first slash is checked if it is recog?nized as a hostname[:port] (i.e., it contains either a . or a :, or the component is ex?actly localhost). If the first component of name is not recognized as a hostname[:port], name is treated as docker.io/name.

docker-archive:path[:{docker-reference|@source-index}]

An image is stored in the docker-save(1) formatted file. docker-reference must not con?tain a digest. Alternatively, for reading archives, @source-index is a zero-based index in archive manifest (to access untagged images). If neither docker-reference nor @_source_index is specified when reading an archive, the archive must contain exactly one image.

It is further possible to copy data to stdin by specifying docker-archive:/dev/stdin but note that the used file must be seekable.

docker-daemon:docker-reference|algo:digest

An image stored in the docker daemon's internal storage. The image must be specified as a docker-reference or in an alternative algo:digest format when being used as an image source. The algo:digest refers to the image ID reported by docker-inspect(1).

oci:path[:tag]

An image compliant with the "Open Container Image Layout Specification" at path. Using a tag is optional and allows for storing multiple images at the same path.

oci-archive:path[:tag]

An image compliant with the "Open Container Image Layout Specification" stored as a tar(1) archive at path.

ostree:docker-reference[@/absolute/repo/path]

An image in the local ostree(1) repository. /absolute/repo/path defaults to /ostree/repo.

Examples

The following examples demonstrate how some of the containers transports can be used. The examples use skopeo-copy(1) for copying container images.

Copying an image from one registry to another:

```
$ skopeo copy docker://docker.io/library/alpine:latest docker://localhost:5000/alpine:latest
```

Copying an image from a running Docker daemon to a directory in the OCI layout:

```
$ mkdir alpine-oci

$ skopeo copy docker-daemon:alpine:latest oci:alpine-oci

$ tree alpine-oci

test-oci/

??? blobs

??? ??? sha256

???     ??? 83ef92b73cf4595aa7fe214ec6747228283d585f373d8f6bc08d66bebab531b7

???     ??? 9a6259e911dcd0a53535a25a9760ad8f2eded3528e0ad5604c4488624795cecc

???     ??? ff8df268d29ccbe81cdf0a173076dcfbbea4bb2b6df1dd26766a73cb7b4ae6f7

??? index.json

??? oci-layout

2 directories, 5 files
```

Copying an image from a registry to the local storage:

```
$ skopeo copy docker://docker.io/library/alpine:latest containers-storage:alpine:latest
```

SEE ALSO

docker-login(1), docker-save(1), ostree(1), podman-login(1), skopeo-copy(1), skopeo-in?spect(1), tar(1), container-registries.conf(5), containers-storage.conf(5)

AUTHORS

Miloslav Trma? mitr@redhat.com ?mailto:mitr@redhat.com? Valentin Rothberg rothberg@red?hat.com ?mailto:rothberg@redhat.com?