Rocky Enterprise Linux 9.2 Manual Pages on command 'podman-push.1'

$ man podman-push.1

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

NAME

podman-push - Push an image, manifest list or image index from local storage to elsewhere

SYNOPSIS

podman push [options] image [destination]

podman image push [options] image [destination]

DESCRIPTION

Pushes an image, manifest list or image index from local storage to a specified destination. Push is mainly used to push images to reg? istries, however podman push can be used to save images to tarballs and directories using the following transports: dir:, docker-archive:, docker-daemon: and oci-archive:.

Image storage

Images are pushed from those stored in local image storage.

DESTINATION

DESTINATION is the location the container image is pushed to. It sup? ports all transports from containers-transports(5). If no transport is

specified, the docker (i.e., container registry) transport is used.

For remote clients, including Mac and Windows (excluding WSL2) ma?

chines, docker is the only supported transport.

```
# Push to a container registry
$ podman push quay.io/podman/stable
# Push to a container registry via the docker transport
$ podman push docker://quay.io/podman/stable
# Push to a container registry with another tag
$ podman push myimage quay.io/username/myimage
# Push to a local directory
$ podman push myimage dir:/tmp/myimage
# Push to a tarball in the docker-archive format
$ podman push myimage docker-archive:/tmp/myimage
# Push to a local docker daemon
$ sudo podman push myimage docker-daemon:docker.io/library/myimage:33
# Push to a tarball in the OCI format
$ podman push myimage oci-archive:/tmp/myimage
```

OPTIONS

--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)

--compress

Compress tarball image layers when pushing to a directory using the

'dir' transport. (default is same compression type, compressed or un?
compressed, as source)

Note: This flag can only be set when using the dir transport

--compression-format=gzip | zstd | zstd:chunked

Specifies the compression format to use. Supported values are: gzip, zstd and zstd:chunked. The default is gzip unless overridden in the containers.conf file.

--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.

--digestfile=Digestfile

After copying the image, write the digest of the resulting image to the file. (This option is not available with the remote Podman client, in? cluding Mac and Windows (excluding WSL2) machines)

--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.

--encrypt-layer=layer(s)

Layer(s) to encrypt: 0-indexed layer indices with support for negative indexing (e.g. 0 is the first layer, -1 is the last layer). If not de? fined, will encrypt all layers if encryption-key flag is specified.

--encryption-key=key

The [protocol:keyfile] specifies the encryption protocol, which can be JWE (RFC7516), PGP (RFC4880), and PKCS7 (RFC2315) and the key material required for image encryption. For instance, jwe:/path/to/key.pem or pgp:admin@example.com or pkcs7:/path/to/x509-file.

--format, -f=format

Manifest Type (oci, v2s2, or v2s1) to use when pushing an image.

--quiet, -q

When writing the output image, suppress progress output

--remove-signatures

Discard any pre-existing signatures in the image.

--sign-by=key

Add a ?simple signing? signature at the destination using the specified key. (This option is not available with the remote Podman client, in? cluding Mac and Windows (excluding WSL2) machines)

--sign-by-sigstore=*param-file***

Add a sigstore signature based on further options specified in a con? tainers sigstore signing parameter file param-file. See containers-sigstore-signing-params.yaml(5) for details about the file format.

--sign-by-sigstore-private-key=path

Add a sigstore signature at the destination using a private key at the specified path. (This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2) machines)

--sign-passphrase-file=path

If signing the image (using either --sign-by or --sign-by-sigstore-pri? vate-key), read the passphrase to use from the specified path.

--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)

EXAMPLE

This example pushes the image specified by the imageID to a local di? rectory in docker format.

podman push imageID dir:/path/to/image

This example pushes the image specified by the imageID to a local di? rectory in oci format.

podman push imageID oci-archive:/path/to/layout:image:tag

This example pushes the image specified by the imageID to a container registry named registry.example.com

podman push imageID docker://registry.example.com/repository:tag

This example pushes the image specified by the imageID to a container registry named registry.example.com and saves the digest in the specified digestfile.

```
# podman push --digestfile=/tmp/mydigest imageID docker://registry.example.com/repository:tag
```

This example pushes the image specified by the imageID and puts it into the local docker container store

```
# podman push imageID docker-daemon:image:tag
```

This example pushes the alpine image to umohnani/alpine on dockerhub and reads the creds from the path given to --authfile

```
# podman push --authfile temp-auths/myauths.json alpine docker://docker.io/umohnani/alpine
Getting image source signatures
Copying blob sha256:5bef08742407efd622d243692b79ba0055383bbce12900324f75e56f589aedb0
 4.03 MB / 4.03 MB [=========================================================] 1s
Copying config sha256:ad4686094d8f0186ec8249fc4917b71faa2c1030d7b5a025c29f26e19d95c156
 1.41 KB / 1.41 KB [=========================================================] 1s
Writing manifest to image destination
Storing signatures
```

This example pushes the rhel7 image to rhel7-dir with the "oci" manifest type

```
# podman push --format oci registry.access.redhat.com/rhel7 dir:rhel7-dir
Getting image source signatures
Copying blob sha256:9cadd93b16ff2a0c51ac967ea2abfadfac50cfa3af8b5bf983d89b8f8647f3e4
 71.41 MB / 71.41 MB [=====================================================] 9s
Copying blob sha256:4aa565ad8b7a87248163ce7dba1dd3894821aac97e846b932ff6b8ef9a8a508a
 1.21 KB / 1.21 KB [=========================================================] 0s
Copying config sha256:f1b09a81455c351eaa484b61aacd048ab613c08e4c5d1da80c4c46301b03cf3b
 3.01 KB / 3.01 KB [=========================================================] 0s
Writing manifest to image destination
Storing signatures
```

SEE ALSO

podman(1), podman-pull(1), podman-login(1), containers-certs.d(5), containers-registries.conf(5), containers-transports(5)