



*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-manifest-push.1' command**

### **\$ man podman-manifest-push.1**

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

#### NAME

podman-manifest-push - Push a manifest list or image index to a registry

#### SYNOPSIS

podman manifest push [options] listnameorindexname [destination]

#### DESCRIPTION

Pushes a manifest list or image index to a registry.

#### RETURN VALUE

The list image's ID and the digest of the image's manifest.

#### OPTIONS

--all

Push the images mentioned in the manifest list or image index, in addition to the list or index itself. (Default true)

--authfile=path

Path of the authentication file. Default is `${XDG_RUNTIME_DIR}/containers/auth.json`, which is set using podman login. If the authorization state is not found there, `$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 authentication file by setting the `REGISTRY_AUTH_FILE` environment variable. This can be done with `export REGISTRY_AUTH_FILE=path`.

--cert-dir=path

Use certificates at path (\*.crt, \*.cert, \*.key) to connect to the registry. (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)

`--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 entered 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, including Mac and Windows (excluding WSL2) machines)

`--format, -f=format`

Manifest list type (oci or v2s2) to use when pushing the list (default is oci).

`--quiet, -q`

When writing the manifest, suppress progress output

`--remove-signatures`

Don't copy signatures when pushing images.

`--rm`

Delete the manifest list or image index from local storage if pushing succeeds.

`--sign-by=fingerprint`

Sign the pushed images with a simple signing signature using the specified key. (This option is not available with the remote Podman client, including Mac and Windows (excluding WSL2) machines)

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

Add a sigstore signature based on further options specified in a con?

containers 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`

Sign the pushed images with a sigstore signature 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-private-key`), read the passphrase to use from the specified path.

`--tls-verify`

Require HTTPS and verify certificates when contacting registries (default: 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)

## DESTINATION

The DESTINATION is a location to store container images

The Image "DESTINATION" uses a "transport":"details" format.

If a transport is not given, podman push will attempt to push to a registry.

Multiple transports are supported:

`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.

```
$ podman manifest push mylist:v1.11 dir:/tmp/mylist
```

`docker://docker-reference`

An image in a registry implementing the "Docker Registry HTTP API V2". By default, uses the authorization state in `$XDG_RUNTIME_DIR/containers/auth.json`, which is set using (podman login). If the authorization state is not found there, `$HOME/.docker/config.json` is checked, which is set using (docker login).

```
$ podman manifest push mylist:v1.11 docker://registry.example.org/mylist:v1.11
```

docker-archive:path[:docker-reference]

An image is stored in the docker save formatted file. docker-reference is only used when creating such a file, and it must not contain a digest.

```
$ podman manifest push mylist:v1.11 docker-archive:/tmp/mylist
docker-daemon:docker-reference
```

An image in docker-reference format stored in the docker daemon internal storage. docker-reference must contain a tag.

```
$ podman manifest push mylist:v1.11 docker-daemon:registry.example.org/mylist:v1.11
oci-archive:path:tag
```

An image tag in a directory compliant with "Open Container Image Layout Specification" at path.

```
$ podman manifest push mylist:v1.11 oci-archive:/tmp/mylist
```

#### EXAMPLE

```
podman manifest push mylist:v1.11 docker://registry.example.org/mylist:v1.11
```

#### SEE ALSO

podman(1), podman-manifest(1)

podman-manifest-push(1)