



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

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

\$ man podman-login.1

podman-login(1)() podman-login(1)()

NAME

podman-login - Login to a container registry

SYNOPSIS

podman login [options] [registry]

DESCRIPTION

podman login logs into a specified registry server with the correct username and password.

If the registry is not specified, the first registry under [registries.search] from reg?

istries.conf will be used. podman login reads in the username and password from STDIN.

The username and password can also be set using the username and password flags. The path

of the authentication file can be specified by the user by setting the authfile flag. The

default path for reading and writing credentials is \${XDG_RUNTIME_DIR}/contain?

ers/auth.json. Podman will use existing credentials if the user does not pass in a user?

name. Podman will first search for the username and password in the \${XDG_RUN?

TIME_DIR}/containers/auth.json, if they are not valid, Podman will then use any existing

credentials found in \$HOME/.docker/config.json. If those credentials are not present,

Podman will create \${XDG_RUNTIME_DIR}/containers/auth.json (if the file does not exist)

and will then store the username and password from STDIN as a base64 encoded string in it.

For more details about format and configurations of the auth.json file, please refer to

containers-auth.json(5)

podman [GLOBAL OPTIONS]

podman login [GLOBAL OPTIONS]

podman login [OPTIONS] [REGISTRY] [GLOBAL OPTIONS]

OPTIONS

`--authfile=path`

Path of the authentication file. Default is `${XDG_RUNTIME_DIR}/containers/auth.json`.

Note: You can also override the default path of the authentication file by setting the `REGISTRY_AUTH_FILE` environment variable. `export REGISTRY_AUTH_FILE=path`

`--cert-dir=path`

Use certificates at path (`*.cert`, `*.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)

`--get-login`

Return the logged-in user for the registry. Return error if no login is found.

`--help, -h`

Print usage statement

`--password, -p=password`

Password for registry

`--password-stdin`

Take the password from stdin

`--tls-verify=true|false`

Require HTTPS and verify certificates when contacting registries (default: true). If explicitly set to true, then TLS verification will be used. If set to false, then 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 `registries.conf`.

`--username, -u=username`

Username for registry

`--verbose, -v`

print detailed information about credential store

EXAMPLES

```
$ podman login docker.io
```

```
Username: umohnani
```

```
Password:
```

```
Login Succeeded!
```

```
$ podman login -u testuser -p testpassword localhost:5000
```

```
Login Succeeded!
```

```
$ podman login --authfile authdir/myauths.json docker.io
```

```
Username: umohnani
```

```
Password:
```

```
Login Succeeded!
```

```
$ podman login --tls-verify=false -u test -p test localhost:5000
```

```
Login Succeeded!
```

```
$ podman login --cert-dir /etc/containers/certs.d/ -u foo -p bar localhost:5000
```

```
Login Succeeded!
```

```
$ podman login -u testuser --password-stdin < testpassword.txt docker.io
```

```
Login Succeeded!
```

```
$ echo $testpassword | podman login -u testuser --password-stdin docker.io
```

```
Login Succeeded!
```

```
$ podman login quay.io --verbose
```

```
Username: myusername
```

```
Password:
```

```
Used: /run/user/1000/containers/auth.json
```

```
Login Succeeded!
```

SEE ALSO

podman(1), podman-logout(1), containers-auth.json(5), containers-certs.d(5), containers-registries.conf(5)

HISTORY

August 2017, Originally compiled by Urvashi Mohnani umohnani@redhat.com
[?mailto:umohnani@redhat.com?](mailto:umohnani@redhat.com)

podman-login(1)()