



**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 'docker-exec.1'***

**\$ man docker-exec.1**

podman-exec(1)() podman-exec(1)()

#### NAME

podman-exec - Execute a command in a running container

#### SYNOPSIS

podman exec [options] container [command [arg ...]]

podman container exec [options] container [command [arg ...]]

#### DESCRIPTION

podman exec executes a command in a running container.

#### OPTIONS

--detach, -d

Start the exec session, but do not attach to it. The command will run in the background and the exec session will be automatically removed when it completes. The podman exec command will print the ID of the exec session and exit immediately after it starts.

--detach-keys=sequence

Specify the key sequence for detaching a container. Format is a single character [a-Z] or one or more ctrl-<value> characters where <value> is one of: a-z, @, ^, [, , or \_. Specifying "" will disable this feature. The default is ctrl-p,ctrl-q.

--env, -e

You may specify arbitrary environment variables that are available for the command to be executed.

--env-file=file

Read in a line delimited file of environment variables.

--interactive, -i=true|false

When set to true, keep stdin open even if not attached. The default is false.

`--latest, -l`

Instead of providing the container name or ID, use the last created container. If you use methods other than Podman to run containers such as CRI-O, the last started container could be from either of those methods. (This option is not available with the remote Podman client)

`--preserve-fds=N`

Pass down to the process N additional file descriptors (in addition to 0, 1, 2). The total FDs will be 3+N.

`--privileged`

Give extended privileges to this container. The default is false.

By default, Podman containers are "unprivileged" and cannot, for example, modify parts of the operating system. This is because by default a container is only allowed limited access to devices. A "privileged" container is given the same access to devices as the user launching the container.

A privileged container turns off the security features that isolate the container from the host. Dropped Capabilities, limited devices, read/only mount points, Apparmor/SELinux separation, and Seccomp filters are all disabled.

Rootless containers cannot have more privileges than the account that launched them.

`--tty, -t`

Allocate a pseudo-TTY.

`--user, -u`

Sets the username or UID used and optionally the groupname or GID for the specified command. The following examples are all valid: `--user [user | user:group | uid | uid:gid | user:gid | uid:group ]`

`--workdir, -w=path`

Working directory inside the container

The default working directory for running binaries within a container is the root directory (/). The image developer can set a different default with the `WORKDIR` instruction, which can be overridden when creating the container.

## Exit Status

The exit code from `podman exec` gives information about why the command within the container failed to run or why it exited. When `podman exec` exits with a non-zero code, the

exit codes follow the chroot standard, see below:

125 The error is with Podman itself

```
$ podman exec --foo ctrID /bin/sh; echo $?
```

```
Error: unknown flag: --foo
```

```
125
```

126 The contained command cannot be invoked

```
$ podman exec ctrID /etc; echo $?
```

```
Error: container_linux.go:346: starting container process caused "exec: \"/etc\": permission denied": OCI runtime error
```

```
126
```

127 The contained command cannot be found

```
$ podman exec ctrID foo; echo $?
```

```
Error: container_linux.go:346: starting container process caused "exec: \"foo\": executable file not found in $PATH":
```

OCI runtime error

```
127
```

Exit code The contained command exit code

```
$ podman exec ctrID /bin/sh -c 'exit 3'; echo $?
```

```
3
```

## EXAMPLES

```
$ podman exec -it ctrID ls
```

```
$ podman exec -it -w /tmp myCtr pwd
```

```
$ podman exec --user root ctrID ls
```

## SEE ALSO

podman(1), podman-run(1)

## HISTORY

December 2017, Originally compiled by Brent Baudebbaude@redhat.com ?mailto:bbaude@red?

hat.com?

podman-exec(1)()