

NAME

docker-container-exec - Execute a command in a running container

SYNOPSIS

docker container exec [OPTIONS] CONTAINER COMMAND [ARG...]

DESCRIPTION

Run a process in a running container.

The command started using **docker exec** will only run while the container's primary process (**PID 1**) is running, and will not be restarted if the container is restarted.

If the container is paused, then the **docker exec** command will wait until the container is unpaused, and then run

CAPABILITIES

privileged gives the process extended Linux capabilities (<https://man7.org/linux/man-pages/man7/capabilities.7.html>) when running in a container.

Without this flag, the process run by **docker exec** in a running container has the same capabilities as the container, which may be limited. Set **--privileged** to give all capabilities to the process.

USER

user sets the username or UID used and optionally the groupname or GID for the specified command.

The followings examples are all valid:

`--user [user | user:group | uid | uid:gid | user:gid | uid:group ]`

Without this argument the command will be run as root in the container.

Exit Status

The exit code from **docker exec** gives information about why the container failed to exec or why it exited. When **docker exec** exits with a non-zero code, the exit codes follow the **chroot** standard, see below:

126 if the contained command cannot be invoked

```
$ docker exec busybox /etc; echo $?
# exec: "/etc": permission denied
docker: Error response from daemon: Contained command could not be invoked
126
```

127 if the contained command cannot be found

```
$ docker exec busybox foo; echo $?
# exec: "foo": executable file not found in $PATH
```

```
docker: Error response from daemon: Contained command not found or does not exist
127
```

Exit code of contained command otherwise

```
$ docker exec busybox /bin/sh -c 'exit 3'
# 3
```

OPTIONS

-d, --detach [=false]	Detached mode: run command in the background
--detach-keys =""	Override the key sequence for detaching a container
-e, --env =	Set environment variables
--env-file =	Read in a file of environment variables
-i, --interactive [=false]	Keep STDIN open even if not attached
--privileged [=false]	Give extended privileges to the command
-t, --tty [=false]	Allocate a pseudo-TTY
-u, --user =""	Username or UID (format: "[:]")
-w, --workdir =""	Working directory inside the container

SEE ALSO

docker-container(1)