docker container exec

By Docker Inc

Published: 2001-01-01 · Archived: 2026-04-05 12:43:17 UTC

Description	Execute a command in a running container
Usage	`docker container exec [OPTIONS] CONTAINER COMMAND [ARG...]`
Aliases **An alias is a short or memorable alternative for a longer command.**	`docker exec`

The `docker exec` command runs a new command in a running container.

The command you specify with `docker exec` only runs while the container's primary process (`PID 1`) is running, and it isn't restarted if the container is restarted.

The command runs in the default working directory of the container.

The command must be an executable. A chained or a quoted command doesn't work.

- This works: `docker exec -it my_container sh -c "echo a && echo b"`
- This doesn't work: `docker exec -it my_container "echo a && echo b"`

Option	Default	Description
`-d, --detach`		Detached mode: run command in the background
`--detach-keys`		Override the key sequence for detaching a container
`-e, --env`		API 1.25+ Set environment variables
`--env-file`		API 1.25+ Read in a file of environment variables
`-i, --interactive`		Keep STDIN open even if not attached
`--privileged`		Give extended privileges to the command
`-t, --tty`		Allocate a pseudo-TTY
`-u, --user`		Username or UID (format: `<name|uid>[:<group|gid>]`)
`-w, --workdir`		API 1.35+ Working directory inside the container

Run `docker exec` on a running container

First, start a container.

This creates and starts a container named `mycontainer` from an `alpine` image with an `sh` shell as its main process. The `-d` option (shorthand for `--detach`) sets the container to run in the background, in detached mode, with a pseudo-TTY attached (`-t`). The `-i` option is set to keep `STDIN` attached (`-i`), which prevents the `sh` process from exiting immediately.

Next, execute a command on the container.

This creates a new file `/tmp/execWorks` inside the running container `mycontainer`, in the background.

Next, execute an interactive `sh` shell on the container.

This starts a new shell session in the container `mycontainer`.

Set environment variables for the exec process (--env, -e)

Next, set environment variables in the current bash session.

The `docker exec` command inherits the environment variables that are set at the time the container is created. Use the `--env` (or the `-e` shorthand) to override global environment variables, or to set additional environment variables for the process started by `docker exec`.

The following example creates a new shell session in the container `mycontainer`, with environment variables `$VAR_A` set to `1`, and `$VAR_B` set to `2`. These environment variables are only valid for the `sh` process started by that `docker exec` command, and aren't available to other processes running inside the container.

Escalate container privileges (--privileged)

See `docker run --privileged`.

Set the working directory for the exec process (--workdir, -w)

By default `docker exec` command runs in the same working directory set when the container was created.

You can specify an alternative working directory for the command to execute using the `--workdir` option (or the `-w` shorthand):

Try to run `docker exec` on a paused container

If the container is paused, then the `docker exec` command fails with an error:
