

**NAME**

`docker-container-attach` - Attach local standard input, output, and error streams to a running container

**SYNOPSIS**

**docker container attach [OPTIONS] CONTAINER**

**DESCRIPTION**

The **docker attach** command allows you to attach to a running container using the container's ID or name, either to view its ongoing output or to control it interactively. You can attach to the same contained process multiple times simultaneously, screen sharing style, or quickly view the progress of your detached process.

To stop a container, use `CTRL-c`. This key sequence sends `SIGKILL` to the container. You can detach from the container (and leave it running) using a configurable key sequence. The default sequence is `CTRL-p CTRL-q`. You configure the key sequence using the **--detach-keys** option or a configuration file. See **config-json(5)** for documentation on using a configuration file.

It is forbidden to redirect the standard input of a `docker attach` command while attaching to a tty-enabled container (i.e.: launched with `-t`).

**Override the detach sequence**

If you want, you can configure an override the Docker key sequence for detach. This is useful if the Docker default sequence conflicts with key sequence you use for other applications. There are two ways to define your own detach key sequence, as a per-container override or as a configuration property on your entire configuration.

To override the sequence for an individual container, use the **--detach-keys="<sequence>"** flag with the `docker attach` command. The format of the `<sequence>` is either a letter [a-Z], or the `ctrl-` combined with any of the following:

- a-z (a single lowercase alpha character)
- @ (at sign)
- [ (left bracket)
- \ (two backward slashes)
- \_ (underscore)
- ^ (caret)

These `a`, `ctrl-a`, `X`, or `ctrl-\` values are all examples of valid key sequences. To configure a different configuration default key sequence for all containers, see **docker(1)**.

**EXAMPLES****Attaching to a container**

In this example the top command is run inside a container, from an image called `fedora`, in detached mode. The ID from the container is passed into the **docker attach** command:

```

$ ID=$(sudo docker run -d fedora /usr/bin/top -b)
$ sudo docker attach $ID
top - 02:05:52 up 3:05, 0 users, load average: 0.01, 0.02, 0.05
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.1%us, 0.2%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 373572k total, 355560k used, 18012k free, 27872k buffers
Swap: 786428k total, 0k used, 786428k free, 221740k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 17200 1116 912 R 0 0.3 0:00.03 top

top - 02:05:55 up 3:05, 0 users, load average: 0.01, 0.02, 0.05
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.2%sy, 0.0%ni, 99.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 373572k total, 355244k used, 18328k free, 27872k buffers
Swap: 786428k total, 0k used, 786428k free, 221776k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 17208 1144 932 R 0 0.3 0:00.03 top

```

**OPTIONS**

<b>--detach-keys=""</b>	Override the key sequence for detaching a container
<b>-h, --help[=false]</b>	help for attach
<b>--no-stdin[=false]</b>	Do not attach STDIN
<b>--sig-proxy[=true]</b>	Proxy all received signals to the process

**SEE ALSO****docker-container(1)**