

Attaching to your container

If you want to go into a container again to execute something you have two options:

- `attach` Attach to the specified process running on the container. In our example it is the nginx server.
- `exec` Executing another process inside the container. This could be a shell, or a script of some sort.

NOTE: When you attach to an already started container, you cannot exit normally without killing the container unless you issue the attach command like this: `docker container attach --sig-proxy=false CONTAINER` or by issuing `Ctrl+p Ctrl+q` If you issue the `exec` command, you can stop it by `Ctrl+d` or detach yourself by `Ctrl+p Ctrl+q`

First, start up an Nginx container:

```
docker container run -d -p 8000:80 nginx
661c6dd59d78b97f8142d67eff6b1d58fbbd42247900241e08f46abdbad19f06
```

Try to attach to the container. Exit it, and browse the webpage again to acknowledge it is gone.

Step into the container by executing a bash inside the container:

```
docker container exec -it CONTAINER bash
```

Inside, we want to run a longer running process, like pinging itself 100 times. Because containers only have the bare minimum installed, we need to first install ping, and then use it:

```
apt-get update && apt-get install iputils-ping -y
ping 127.0.0.1 -c 100 > /tmp/ping
```

Then detach from the container with `Ctrl+p Ctrl+q` and run the following:

```
docker container exec -it container bash
tail -f /tmp/ping
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.156 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.183 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.158 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.139 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.156 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.139 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.118 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.155 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.149 ms
```

```
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.202 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.154 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.151 ms
64 bytes from 127.0.0.1: icmp_seq=14 ttl=64 time=0.144 ms
64 bytes from 127.0.0.1: icmp_seq=15 ttl=64 time=0.148 ms
64 bytes from 127.0.0.1: icmp_seq=16 ttl=64 time=0.163 ms
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.146 ms
64 bytes from 127.0.0.1: icmp_seq=18 ttl=64 time=0.156 ms
64 bytes from 127.0.0.1: icmp_seq=19 ttl=64 time=0.151 ms
64 bytes from 127.0.0.1: icmp_seq=20 ttl=64 time=0.147 ms
64 bytes from 127.0.0.1: icmp_seq=21 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_seq=22 ttl=64 time=0.156 ms
64 bytes from 127.0.0.1: icmp_seq=23 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=24 ttl=64 time=0.139 ms
64 bytes from 127.0.0.1: icmp_seq=25 ttl=64 time=0.153 ms
64 bytes from 127.0.0.1: icmp_seq=26 ttl=64 time=0.158 ms
64 bytes from 127.0.0.1: icmp_seq=27 ttl=64 time=0.198 ms
```

Here you see that the ping process started in another shell is still running and producing this logfile. Just stop the process with a `Ctrl+d`.

summary

You have tried to `attach` directly to the primary process of a container, as well as starting a new process by the `exec` command. You have also seen how to break out of a container either by terminating the process by `Ctrl+d`, or by detaching from the process by `Ctrl+p Ctrl+q`.