

In the previous scenarios we've launched some containers, but like any process, containers can crash. This scenario will explore how you can keep containers live and automatically restart them if the crash unexpectedly.

Step 1 - Stop On Fail

Docker considers any containers to exit with a non-zero exit code to have crashed. By default a crashed container will remain stopped.

Example

We've created a special container which outputs a message and then exits with code 1 to simulate a crash.

You can launch an instance using `docker run -d --name restart-default scrapbook/docker-restart-example`

If you list all the containers, including stopped, you will see the container has crashed `docker ps -a`

While the logs will output our message, which in real-life would hopefully indicate information to help us diagnose the issue.

`docker logs restart-default`

```
$ docker run -d --name restart-default scrapbook/docker-restart-example
8273354a807c4299258c63a2fc7fc405e2a713fa2d3b4b89c589b3b1a3b88d24
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
8273354a807c        scrapbook/docker-restart-example   "/bin/sh -c './launch..."   4 seconds ago       Exited (1) 2 seconds ago              restart-default
$ docker logs restart-default
Mon Apr 17 09:17:29 UTC 2023 Booting up...
$
```

Step 2 - Restart On Fail

Depending on your scenario, restarting a failed process might correct the problem. Docker can automatically retry to launch the Docker a specific number of times before it stops trying.

Example

The option `--restart=on-failure:#` allows you to say how many times Docker should try again. In the example below, Docker will restart the container three times before stopping.

`docker run -d --name restart-3 --restart=on-failure:3 scrapbook/docker-restart-example`

As we can see from the logs, it was launched on three occasions.

`docker logs restart-3`

```
Mon Apr 17 09:17:29 UTC 2023 Booting up...
$ docker run -d --name restart-3 --restart=on-failure:3 scrapbook/docker-restart-example
43d98a6aea3a60091a8c4c9ce94af03124ce95602863b2332015990c2b95cc9a
$ docker logs restart-1
Error: No such container: restart-1
$ docker logs restart-3
Mon Apr 17 09:19:57 UTC 2023 Booting up...
Mon Apr 17 09:19:58 UTC 2023 Booting up...
Mon Apr 17 09:20:00 UTC 2023 Booting up...
Mon Apr 17 09:20:01 UTC 2023 Booting up...
$
```

Step 3 - Always Restart

Finally Docker can always restart a failed container, in this case, Docker will keep trying until the container it is explicitly told to stop.

Example

Use the always flag to automatically restart the container when is crashes for example

`docker run -d --name restart-always --restart=always scrapbook/docker-restart-example`

You can view the restart attempting via the log

`docker logs restart-always`

```
Mon Apr 17 09:20:01 UTC 2023 Booting up...
$ docker run -d --name restart-always --restart=always scrapbook/docker-restart-example
cc0bcb78725a47f16bc27e6a3f16e89ec2cfb990feb3ef64ce51b3fe48e0caf9
$ docker logs restart-always
Mon Apr 17 09:24:11 UTC 2023 Booting up...
Mon Apr 17 09:24:12 UTC 2023 Booting up...
Mon Apr 17 09:24:13 UTC 2023 Booting up...
Mon Apr 17 09:24:15 UTC 2023 Booting up...
Mon Apr 17 09:24:17 UTC 2023 Booting up...
$
```

This scenario has introduced how you can keep alive containers even after they crash.