

Title: How to stop all containers when one container stops with docker-compose?

Post Body:

Up until recently, when one was doing `docker-compose up` for a bunch of containers and one of the started containers stopped, all of the containers were stopped. This is not the case anymore since <https://github.com/docker/compose/issues/741> and this is a really annoying for us: We use docker-compose to run selenium tests which means starting application server, starting selenium hub + nodes, starting tests driver, then exiting when tests driver stops.

Is there a way to get back old behaviour?

Accepted Answer:

You can use:

```
docker-compose up --abort-on-container-exit
```

Which will stop all containers if one of your containers stops

Highest Rated Answer:

In your docker compose file, setup your *test driver* container to depend on other containers (with `depends_on` parameter). Your docker compose file should look like this:

```
services:  application_server:      ...  selenium:      ...  test_driver:  entry_point: YOUR_TEST_COMMAND  depends_on
```

With dependencies expressed this way, run:

```
docker-compose run test_driver
```

and all the other containers will shut down when the `test_driver` container is finished.

This solution is an alternative to the `docker-compose up --abort-on-container-exit` [answer](#). The latter will also shut down all other containers if any of them exits (not only the *test driver*). It depends on your use case which one is more adequate.