Title: Docker-compose exit code is 137 when there is no OOM exception

Post Body:

When I run the following command, I expect the exit code to be 0 since my `combined` container runs a test that successfully exits with an exit code of 0.

```
docker-compose up --build --exit-code-from combined
```

Unfortunately, I consistently receive an exit code of 137 even when the tests in my `combined` container run successfully and I exit that container with an exit code of 0 (more details on how that happens are specified below).

Below is my docker-compose version:

```
docker-compose version 1.25.0, build 0a186604
```

According to this post, the exit code of 137 can be due to two main issues.

1. The container received a `docker stop` and the app is not gracefully handling SIGTERM
2. The container has run out of memory (OOM).

I know the 137 exit code is not because my container has run out of memory. When I run `docker inspect <container-id>`, I can see that 'OOMKilled' is false as shown in the snippet below. I also have 6GB of memory allocated to the Docker Engine which is plenty for my application.

```
[     {         'Id': 'db4a48c8e4bab69edff479b59d7697362762a8083db2b2088c58945fcb005625',         'Created': '2019-12-12T01:43
```

My container doesn't exit from a `docker stop` so I don't think the first reason is relevant to my situation either.

How my Docker containers are set up

I have two Docker containers:

1. **b-db** - contains my database
2. **b-combined** - contains my web application and a series of tests, which run once the container is up and running.

I'm using a docker-compose.yml file to start both containers.

```
version: '3' services:     db:         build:             context: .             dockerfile: ./docker/db/Dockerfile         co
```

Below is the Dockerfile for the `combined` service in `docker-compose.yml`.

```
FROM cypress/included:3.4.1  WORKDIR /usr/src/app  COPY package*.json ./  RUN npm install  COPY . .  EXPOSE 5000  RUN npm inst
```

Below is what is in my `init.sh` file.

```
#!/bin/bash # Start front end server history-server dist -p 8080 & front_pid=$!  # Start back end server that interacts with D
```

Below is the Dockerfile for my `db` service. All its doing is copying some local data into the Docker container and then initialising the database with this data.

```
FROM  mongo:3.6.14-xenial  COPY ./dump/ /tmp/dump/  COPY mongo_restore.sh /docker-entrypoint-initdb.d/  RUN chmod 777 /docker-
```

Below is what is in `mongo_restore.sh`.

```
#!/bin/bash # Creates db using copied data mongorestore /tmp/dump
```

Below are the last few lines of output when I run `docker-compose up --build --exit-code-from combined; echo $?`.

```
... b-combined | user disconnected b-combined | Mongoose disconnected b-combined | Mongoose disconnected through Heroku app sh
```

What is confusing as you can see above, is that the test and script ended with exit code of 0 since all my tests passed successfully but the container still exited with an exit code of 137.

What is even more confusing is that when I comment out the following line (which runs my Cypress integration tests) from my `init.sh` file, the container exits with a 0 exit code as shown below.

```
NODE_ENV=test $(npm bin)/cypress run --config video=false --browser chrome
```

Below is the output I receive when I comment out / remove the above line from `init.sh`, which is a command that runs my Cypress integration tests.

```
... b-combined | TEST ENDED WITH EXIT CODE OF: 0 ========================== b-combined | EXITING SCRIPT WITH EXIT CODE OF: 0
```

How do I get docker-compose to return me a zero exit code when my tests run successfully and a non-zero exit code when they fail?

EDIT:

After running the following docker-compose command in debug mode, I noticed that b-db seems to have some trouble shutting down and potentially is receiving a SIGKILL signal from Docker because of that.

```
docker-compose --log-level DEBUG up --build --exit-code-from combined; echo $?
```

Is this indeed the case according to the following output?

```
... b-combined exited with code 0 Aborting on container exit... http://localhost:None 'GET /v1.25/containers/196f3e622847b4c4c
```

Accepted Answer:

The error message strikes me as: `Aborting on container exit...`

From [docker-compose docs]:

> **--abort-on-container-exit** Stops all containers if any container was stopped.

Are you running docker-compose with this flag? If that is the case, think about what it means.

Once `b-combined` is finished, it simply exits. That means, container `b-db` will be forced to stop as well. Even though `b-combined` returned with exit code 0, `b-db` forced shutdown was likely not handled gracefully by mongodb.

EDIT: I just realized you have `--exit-code-from` in the command line. That implies `--abort-on-container-exit`.

Solution: `b-db` needs more time to exit gracefully. Using `docker-compose up --timeout 600` avoids the error.

Highest Rated Answer:

Docker exit code 137 may imply Docker doesn't have enough RAM to finish the work.

Unfortunately Docker consumes a lot of RAM.

Go to Docker Desktop app > Preferences > Resources > Advanced and increase the MEMORY - best to double it.