Title: Wait for docker-compose to start before running another process

Post Body:

I have an application that uses docker-compose and maven to run integration tests. This application works locally, when I can control when to execute the commands. I first run `docker-compose up` and I wait until everything is done. Then I open another window, and then I run integration tests against the docker-compose container we had just spun up: `mvn verify -P integration-test -DBASE_URL=http://localhost:8080`

However, when I run in gitlab, I don't know how to wait to run the mvn command until after the container spins up. Does anyone have an idea on how I can do this (keep in mind that I think I must run docker-compose up -d in gitlab, otherwise it blocks the runner). The error logs I am seeing are: `org.springframework.web.client.ResourceAccessException: I/O error on GET request for 'http://localhost:8080/health': Connection refused (Connection refused); nested exception is java.net.ConnectException: Connection refused (Connection refused)`

The full gitlab output can be found here: https://gitlab.com/connorbutch/trading-app/-/jobs/524923294

Thanks, Connor

Accepted Answer: None

Highest Rated Answer:

if `http://localhost:8080/health` is an indication that tests are ready to run, you might consider waiting for it to succeed before proceeding. It would be natural to execute docker-compose and mvn from a shell, so I'll provide bash here.

```
wait_for_health() {   for i in $(seq 1 5); do      if curl -s 'http://localhost:8080/health'; then return 0; fi     sleep 2
```

if you're thinking about cleanup, bash can run a command at exit, maybe something like:

```
trap 'docker-compose down' EXIT
```

Finally, remember to run your scripts with `-e` set. You can put it at the shebang `#!/bin/bash -e` or `set -e`, but if you do not set `-e`, failed commands will, counterintuitively, not cause the script to fail. This is typically not what you want.

As an alternative, you can run `mvn` from within docker-compose. You'll have to describe your maven environment and ancillaries in docker, but then you can add a container that depends on the web container. From what I can tell, it's up to you to figure out how to *not* execute `mvn` locally in that case (Can you define optional docker-compose services? seems like a good start), but it might help handle the container dependencies for you.