Exercise 03 - Martin Schneglberger - S1810455022

First try:

```
language: go

services:
  - postgresql

script:
  - go test -v
```

Resulted in:

```
2020/05/27 13:19:16 pq: role "password=" does not exist
exit status 1
```

Might forgot to specify go language version, but this should not be the problem

https://docs.travis-ci.com/user/database-setup/#postgresql

=> Apparently the database and the user has to be setup beforehand

Also, this reads like a password error. After checking my EX02 again, I noticed that I used *secret* as a password, and of course travis cannot know this when setting up the service.
Thus, I have to somehow set envionment variables for travis! But how should this be possible if no credentials should be commited? It would be stupid to set them up as envionment variables for the project just to have them as plain text in the yml again.

Solution: https://docs.travis-ci.com/user/environment-variables/

I will just handle all my environment variables as being sensitive and thus will encrypt them all.
As they might be different for every branch, I will add them to my repository settings as described here:

https://docs.travis-ci.com/user/environment-variables/#defining-variables-in-repository-settings

Thus, new yml:

```
language: go

go:
  - 1.14.x

services:
  - postgresql
```

```
script:
  - go test -v
```

And environment settings:

Environment Variables

Customize your build using environment variables. For secure tips on generating private keys read our documentation

APP_DB_NAME	●●●●●●●●●●●●●●●	Available to all branches	🗑
APP_DB_PASSWORD	●●●●●●●●●●●●●●●	Available to all branches	🗑
APP_DB_USERNAME	●●●●●●●●●●●●●●●	Available to all branches	🗑

I like how the environment is recreated from scratch (thus pulling all the packages) upon each build (and each running in a fresh VM and cloning the repo into it)

AAAAANNNND: Got Ya!

```
▶ 185  $ go env
▶ 219  $ travis_install_go_dependencies 1.14.x -v
  227  $ go test -v
  228  === RUN   TestEmptyTable
  229  --- PASS: TestEmptyTable (0.00s)
  230  === RUN   TestGetNonExistentProduct
  231  --- PASS: TestGetNonExistentProduct (0.00s)
  232  === RUN   TestCreateProduct
  233  --- PASS: TestCreateProduct (0.00s)
  234  === RUN   TestGetProduct
  235  --- PASS: TestGetProduct (0.00s)
  236  === RUN   TestUpdateProduct
  237  --- PASS: TestUpdateProduct (0.00s)
  238  === RUN   TestDeleteProduct
  239  --- PASS: TestDeleteProduct (0.00s)
  240  === RUN   TestOrder66
  241  --- PASS: TestOrder66 (0.00s)
  242  === RUN   TestRedirectWebstore
  243  --- PASS: TestRedirectWebstore (0.00s)
  244  PASS
  245  ok      github.com/schnegl/go_product_api.git    0.029s
  246  The command "go test -v" exited with 0.
  247
  248
  249  Done. Your build exited with 0.
```