Automated Continious Integration with Travis

John Ghatas

Contents

1 Project definition

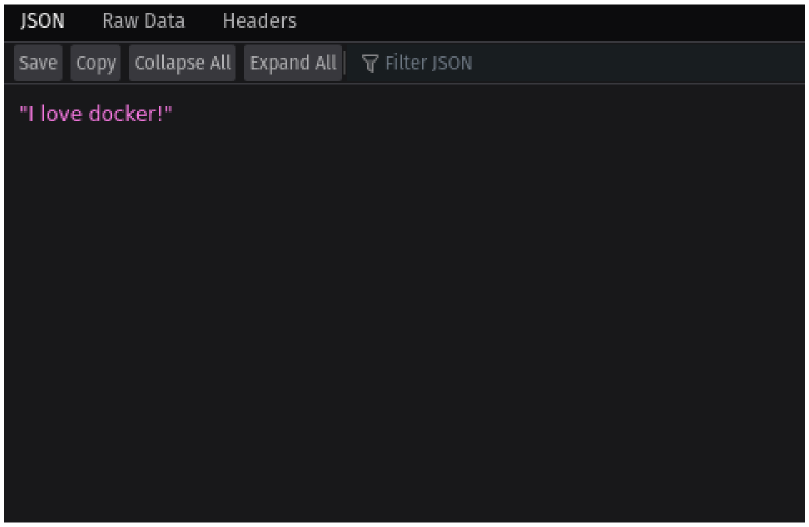
Goal This project was pulled from a Lynda tutorial teaching the basics of Docker, the Docker container created runs an express backend using MongoDB as the database server.

2 Running the application

The first order of business was to ensure that the application was able to run locally before deploying it in a Docker container. To do this we ran the following commands in the main directory:

```
$ npm install
$ npm start
```

To start the backend locally **MongoDB** is required to have a local install. An example of the backend is shown in **Figure 1**

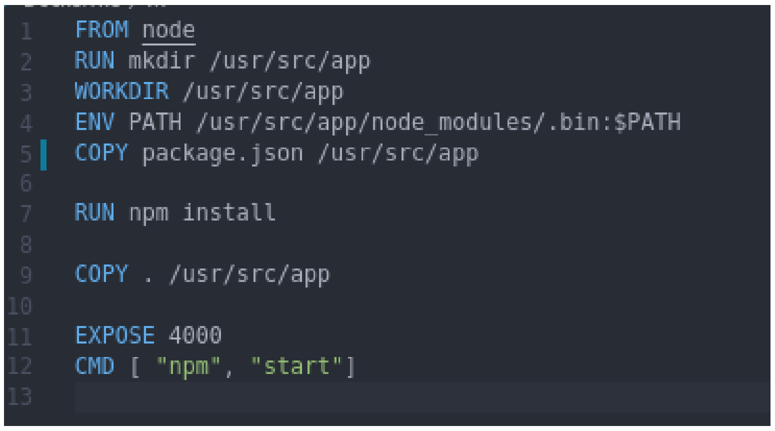


```
"I love docker!"
```

Fig. 1: The root route of the backend running

3 Exporting to docker

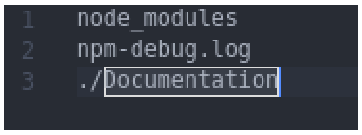
The next step was to export the image to docker locally, to ensure the setup was running before deploying the automation of the Docker image to the TravisCI servers. To build a Docker image, I had to create a Dockerfile specifying the steps that needed to be taken to deploy the Express backend in a Docker container. The Dockerfile is shown in **Figure 2**.



```
FROM node
RUN mkdir /usr/src/app
WORKDIR /usr/src/app
ENV PATH /usr/src/app/node_modules/.bin:$PATH
COPY package.json /usr/src/app

RUN npm install

COPY . /usr/src/app

EXPOSE 4000
CMD [ "npm", "start"]
```

Fig. 2: The dockerfile containing the instructions for building the container

To prevent any unnecessary files being transferred to the Docker container, a **.dockerignore** file is defined as shown in **Figure 3**.



```
node_modules
npm-debug.log
./Documentation
```

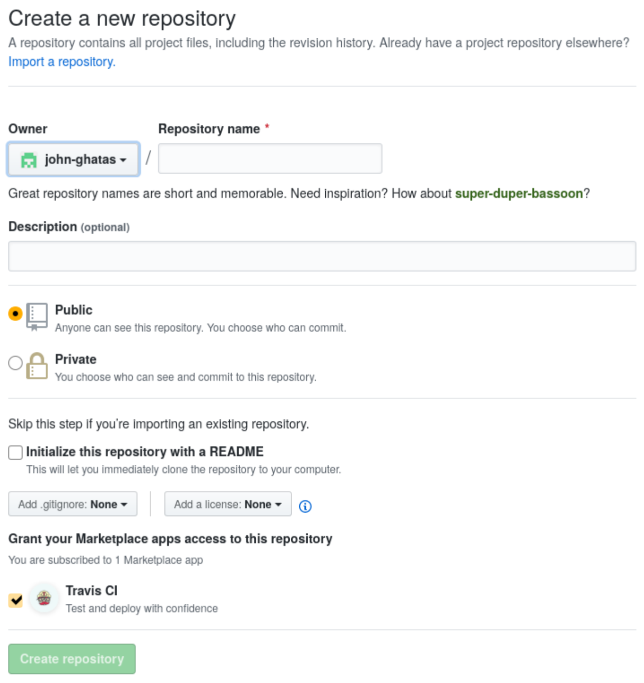
Fig. 3: The ignored files and folders when building the Docker container

We ignored the Node Modules folder and the debug log from the Node Package Manager, thus reducing the total size of the container. Now we could build the image with following command (assuming you are in the root folder of the repository):

```
$ docker build . -t [domain]/[name]
$ docker run -p [local port]:[docker port]
-d [domain]/[name]
```

4 Configuring TravisCI

Before continuing, we have to create the repository, on the homepage of GitHub go to the left hand panel and click on ”New”. This button sends you to the page seen in **Figure 4**.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: john-ghatas / Repository name *

Great repository names are short and memorable. Need inspiration? How about super-duper-bassoon?

Description (optional)

● Public — Anyone can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README — This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None

Grant your Marketplace apps access to this repository

You are subscribed to 1 Marketplace app

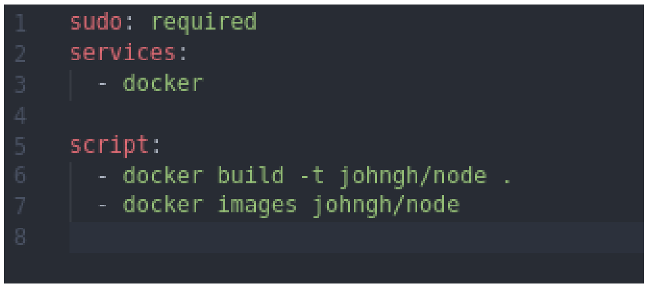
✓ Travis CI — Test and deploy with confidence

Fig. 4: The configuration of the GitHub repository when created

Fill in the repository name, keep the repository on **public** because Travis will be on a limited plan otherwise. Make sure the Travis CI box is checked, and init the repo with a README if you prefer to do so.

4.1 Deploying TravisCI

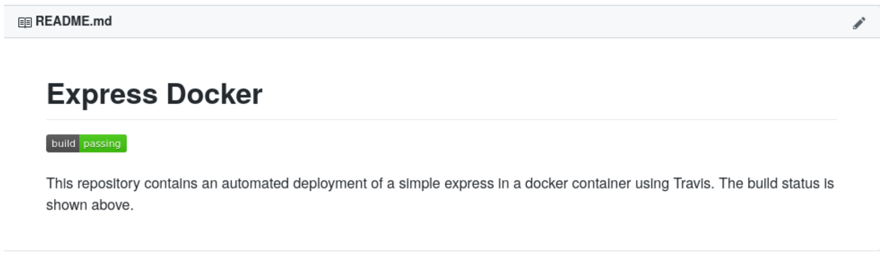
Now that I ensured the backend runs locally, we were ready to automate the building of the docker image on Travis. To configure Travis, a **.travis.yml** file was created to specify the instructions for the CI tool. In **Figure 5** the configuration is shown of the travis file.



```
sudo: required
services:
  - docker

script:
  - docker build -t johngh/node .
  - docker images johngh/node
```

Fig. 5: The configuration file of for the CI tool

The configuration file will be picked by the plugin TravisCI, that I installed on the repository automating the build. After the build is done the status will be updated in the README of the repository as can be seen in **Figure 6**.



README.md

Express Docker

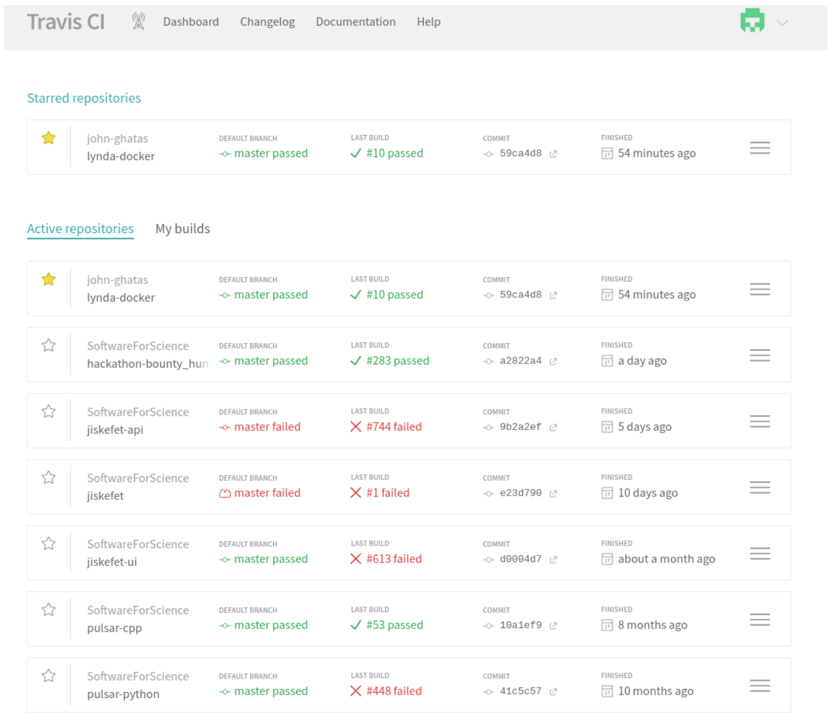
build passing

This repository contains an automated deployment of a simple express in a docker container using Travis. The build status is shown above.

Fig. 6: The build status of the build of the Docker container

4.2 Adding the indicator

As seen in **Figure 6** and indicator of the build status was added indicating whether the build of the Docker container passed or not. To get this container I had to login to my GitHub account linked to TravisCI at the portal, after which I could see my projects linked to my GitHub shown in **Figure 7**.



Starred repositories

	DEFAULT BRANCH	LAST BUILD	COMMIT	FINISHED
john-ghatas lynda-docker	master passed	✓ #10 passed	59ca4d8	54 minutes ago

Active repositories My builds

	DEFAULT BRANCH	LAST BUILD	COMMIT	FINISHED
john-ghatas lynda-docker	master passed	✓ #10 passed	59ca4d8	54 minutes ago
SoftwareForScience hackathon-bounty_hun	master passed	✓ #283 passed	a2822a4	a day ago
SoftwareForScience jiskefet-api	master failed	✗ #744 failed	9b2a2ef	5 days ago
SoftwareForScience jiskefet	master failed	✗ #1 failed	e23d790	10 days ago
SoftwareForScience jiskefet-ui	master passed	✗ #613 failed	d0004d7	about a month ago
SoftwareForScience pulsar-cpp	master passed	✓ #53 passed	10a1ef9	8 months ago
SoftwareForScience pulsar-python	master passed	✗ #448 failed	41c5c57	10 months ago

Fig. 7: The portal with projects linked to my GitHub repositories

The project I configured for building the docker container is called "lynda-docker" which I favourited in the screenshot in **Figure 7**. When I click on the project I see the status of the build in the header as shown in **Figure 8**.

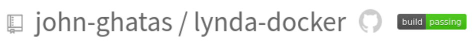
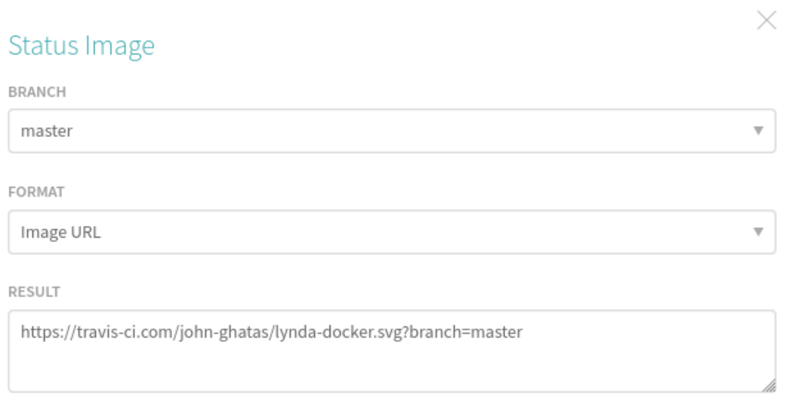


Fig. 8: The status of the project deploying Docker containers

Click on the build status, and you are presented with a screen options to embed the status. The default option an Image URL as seen in **Figure 9**, click on the format and change it to Markdown, copy the result into the README.MD of the repo and the build status is embedded.


Status Image

BRANCH

master

FORMAT

Image URL

RESULT

https://travis-ci.com/john-ghatas/lynda-docker.svg?branch=master

Fig. 9: The embed options for the result