WEB APPLICATIONS DEPLOYMENT

TERM 2

ASSESSABLE ACTIVITY

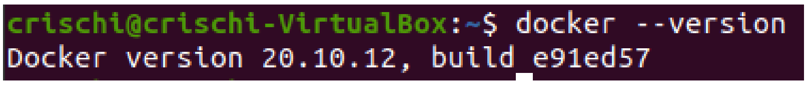
2021-2022

CRISTINA CHIARRI

Sumario

Part 1: Creating a Docker image of a Node.js application

We are going to work from an Ubuntu virtual machine as a private server machine, with a Docker engine installed.
We can confirm that docker is installed using **$ docker --version**



```
crischi@crischi-VirtualBox:~$ docker --version
Docker version 20.10.12, build e91ed57
```

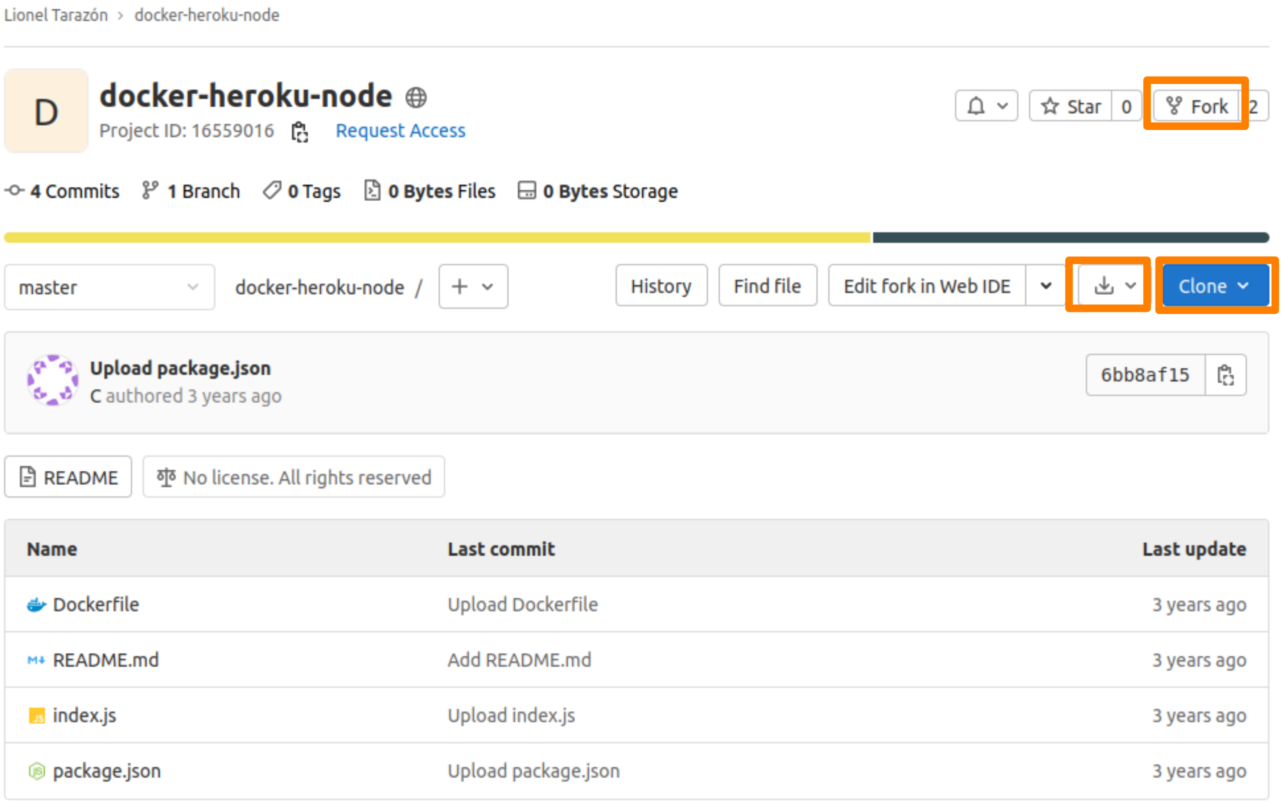
Getting the node.js application from GitLab

We need to clone the content to our web repository from the web repository
https://gitlab.com/lionel_ceedcv/docker-heroku-node

We can do this in different ways:

- cloning the repository into a local repository
- downloading the project as a tar.gz
- forking it to our web repository

We are using the last option, the fork. A fork creates a completely independent copy of Git repository in our repository.



Lionel Tarazón › docker-heroku-node

docker-heroku-node

Project ID: 16559016 Request Access

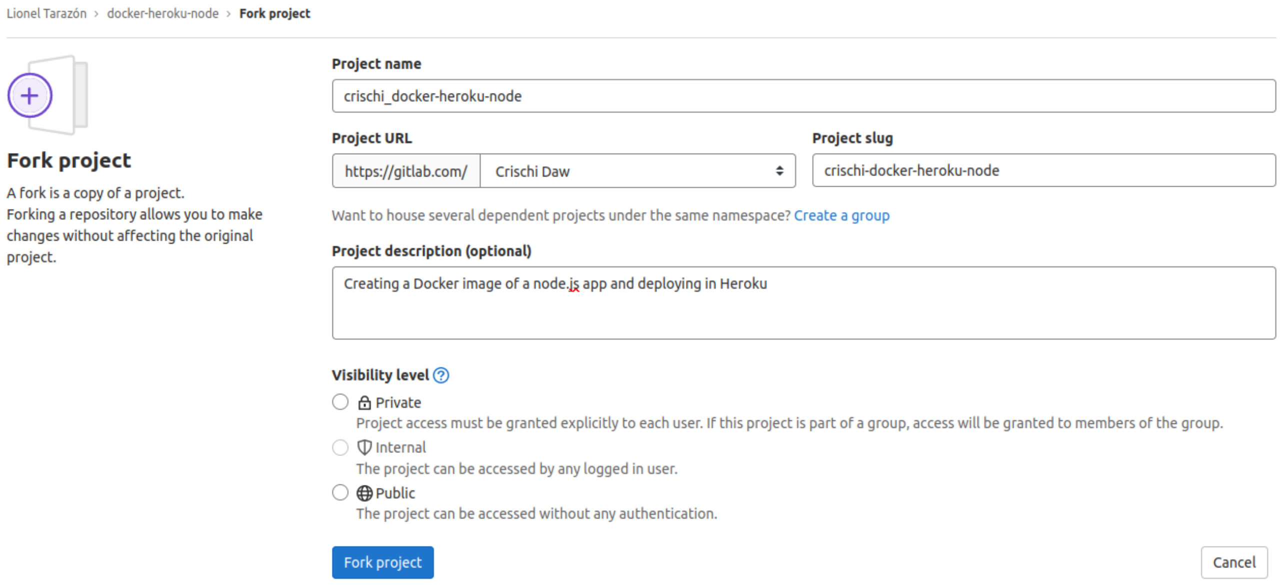
4 Commits 1 Branch 0 Tags 0 Bytes Files 0 Bytes Storage

master docker-heroku-node /

Upload package.json
C authored 3 years ago
6bb8af15

Name	Last commit	Last update
Dockerfile	Upload Dockerfile	3 years ago
README.md	Add README.md	3 years ago
index.js	Upload index.js	3 years ago
package.json	Upload package.json	3 years ago

You can make changes in your own fork and submit them through a merge request to the original repository. In this case, we do not want to merge the changes. It is just a way to create an idenpendent copy of the project.



Lionel Tarazón › docker-heroku-node › **Fork project**

Fork project

A fork is a copy of a project.
Forking a repository allows you to make changes without affecting the original project.

Project name

crischi_docker-heroku-node

Project URL

https://gitlab.com/ | Crischi Daw

Project slug

crischi-docker-heroku-node

Want to house several dependent projects under the same namespace? Create a group

Project description (optional)

Creating a Docker image of a node.js app and deploying in Heroku

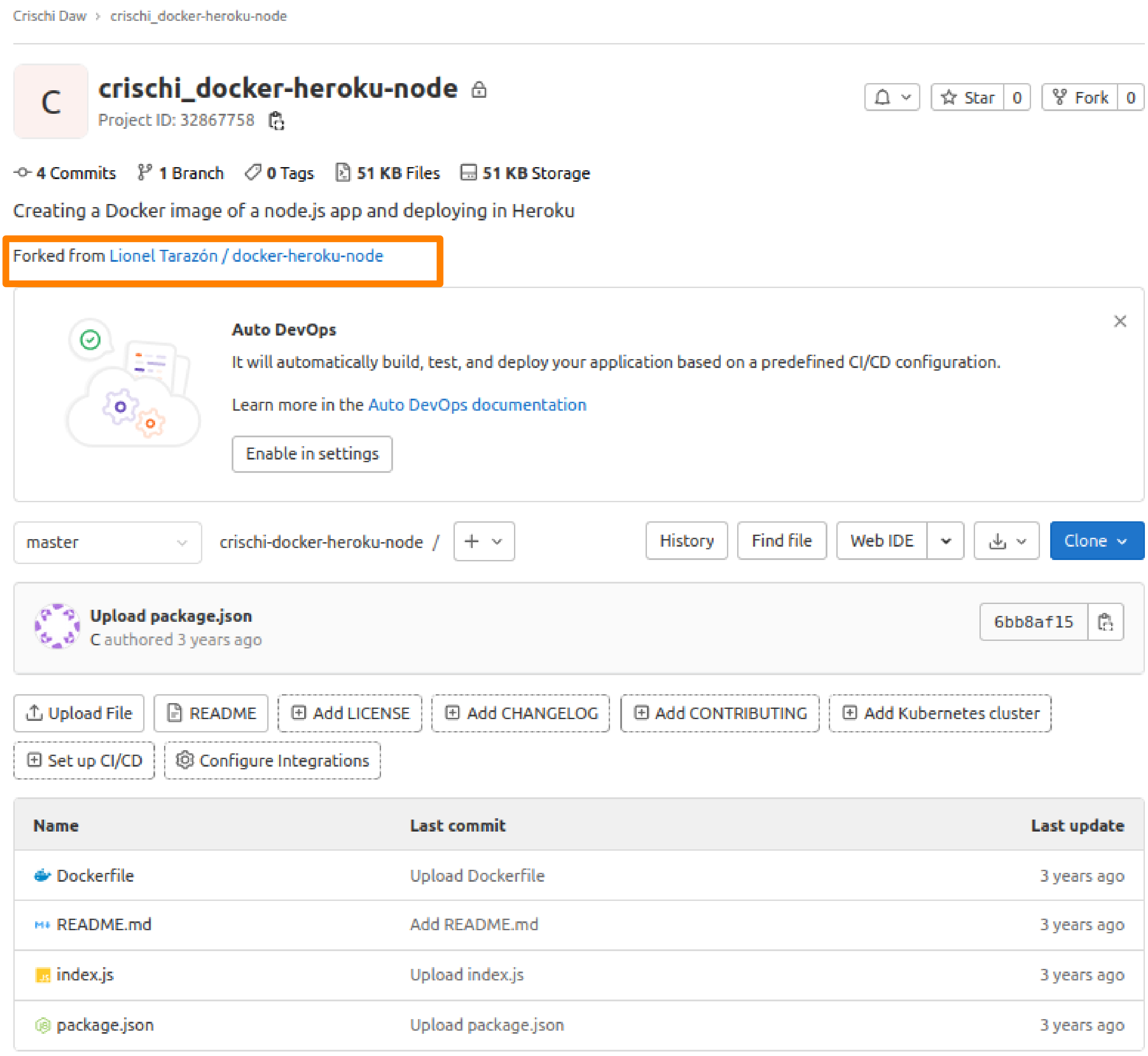
Visibility level

○ Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

○ Internal
The project can be accessed by any logged in user.

○ Public
The project can be accessed without any authentication.

We have now a copy from the original project in our web repository.



Crischi Daw › crischi_docker-heroku-node

crischi_docker-heroku-node

Project ID: 32867758

4 Commits · 1 Branch · 0 Tags · 51 KB Files · 51 KB Storage

Creating a Docker image of a node.js app and deploying in Heroku

Forked from Lionel Tarazón / docker-heroku-node

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the Auto DevOps documentation

Upload package.json
C authored 3 years ago

6bb8af15

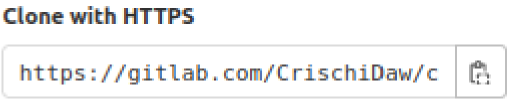
Name	Last commit	Last update
Dockerfile	Upload Dockerfile	3 years ago
README.md	Add README.md	3 years ago
index.js	Upload index.js	3 years ago
package.json	Upload package.json	3 years ago

Creating our local repository

Once we have the original files, we have to clone our web repository to a new local one to use it as worlking repository.

We have created a folder named activity2 to add our local repository.
$ cd activity2

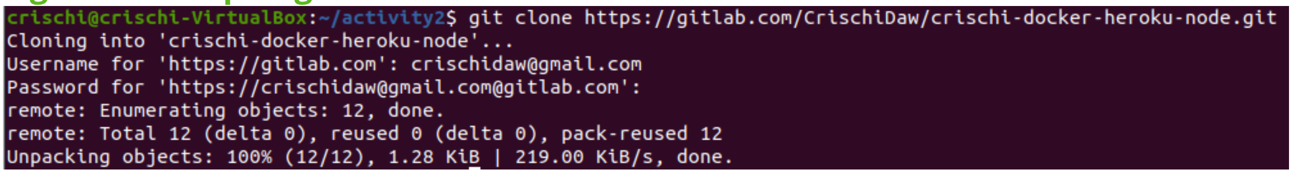
We clone our web repository with **$ git clone <path to our web repository>**
Git clone will create a new local directory for the repository, copy all the contents of the specified repository, create the remote tracked branches, and checkout an initial branch locally.



Clone with HTTPS

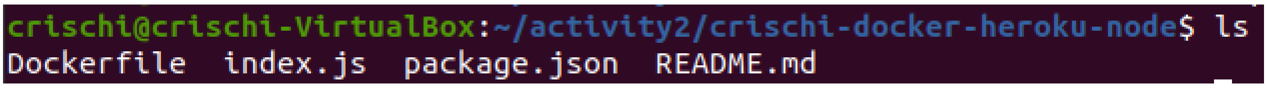
https://gitlab.com/CrischiDaw/c

$ git clone https://gitlab.com/CrischiDaw/crischi-docker-heroku-node



```
crischi@crischi-VirtualBox:~/activity2$ git clone https://gitlab.com/CrischiDaw/crischi-docker-heroku-node.git
Cloning into 'crischi-docker-heroku-node'...
Username for 'https://gitlab.com': crischidaw@gmail.com
Password for 'https://crischidaw@gmail.com@gitlab.com':
remote: Enumerating objects: 12, done.
remote: Total 12 (delta 0), reused 0 (delta 0), pack-reused 12
Unpacking objects: 100% (12/12), 1.28 KiB | 219.00 KiB/s, done.
```

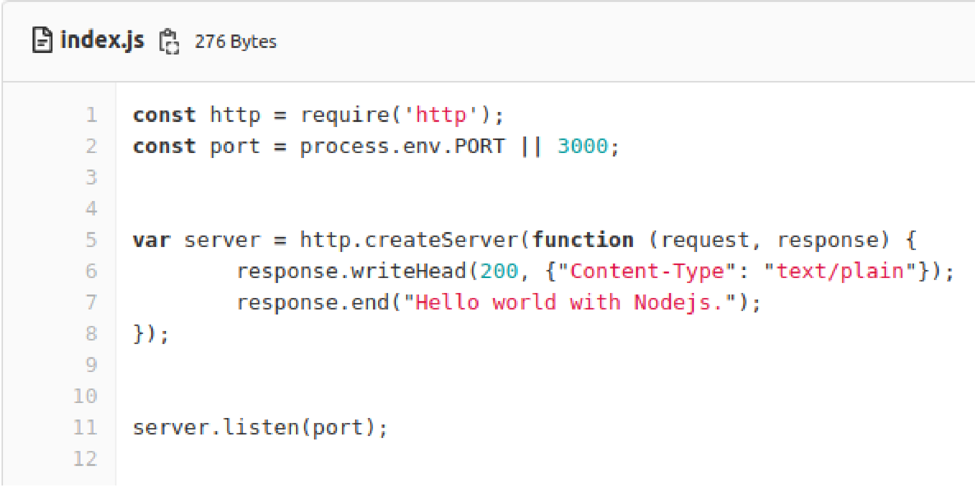
Here we can see the content of the app folder



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ ls
Dockerfile  index.js  package.json  README.md
```

Index.js

This is our main application file.



index.js 276 Bytes

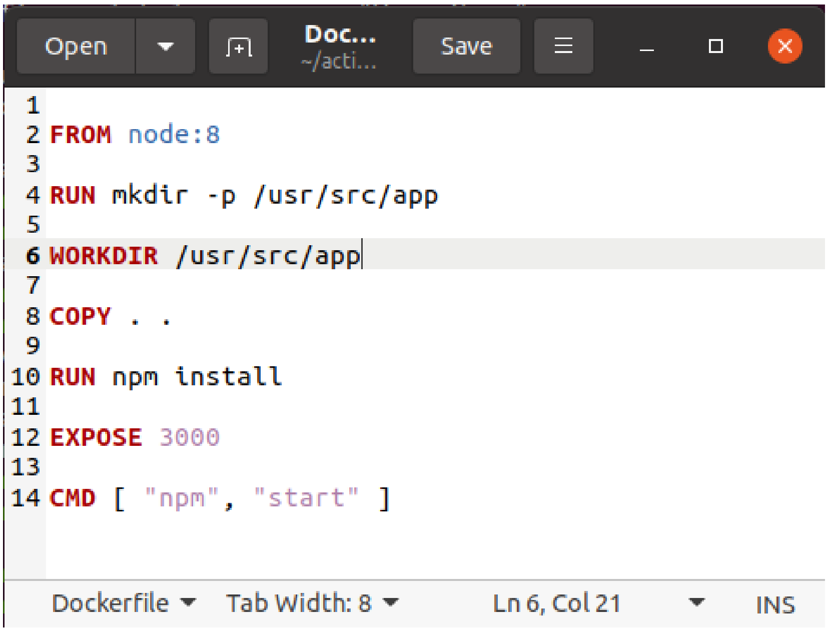
```
const http = require('http');
const port = process.env.PORT || 3000;


var server = http.createServer(function (request, response) {
        response.writeHead(200, {"Content-Type": "text/plain"});
        response.end("Hello world with Nodejs.");
});


server.listen(port);

```

The environment variable **port** is already set. It tells our web server which port to listen on.
process.env.PORT || 3000 means the manually set port number .
3000 is the default port. If it is not manually set, then it will listen on port 3000.
server.listen(port) makes your server able to accept a "which port to listen on" parameter from the environment.

Dockerfile

To create a Docker image, we need a Dockerfile which is a text document that contains all the commands a user could call on the command line to assemble an image.

We already have a Dockerfile in our the gitlab repository.
Here we have the content detail and explanation:

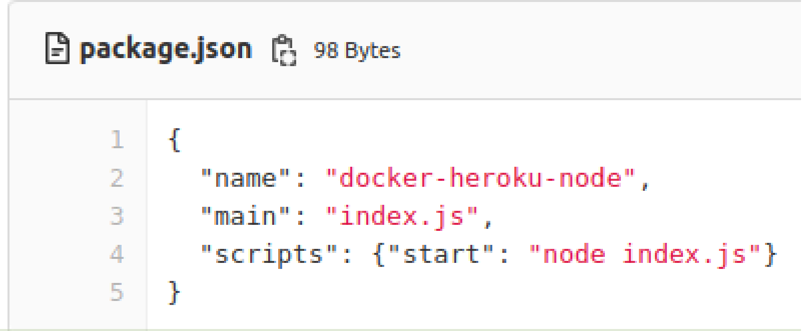


```

FROM node:8

RUN mkdir -p /usr/src/app

WORKDIR /usr/src/app

COPY . .

RUN npm install

EXPOSE 3000

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

FROM node:8	The Docker image will use node:8 from the Docker Hub.
RUN mkdir – p /usr/src/app	Create a new folder for our image
WORKDIR p/usr/src/app	Set the folder as working directory
COPY . .	Copy files or directories from a source to a destination path in the container filesystem.
RUN npm install	Runs the command **npm install**
EXPOSE 3000	Indicates the port where container listens
CMD [“npm”, “start”]	Set the default executable to run

package.json

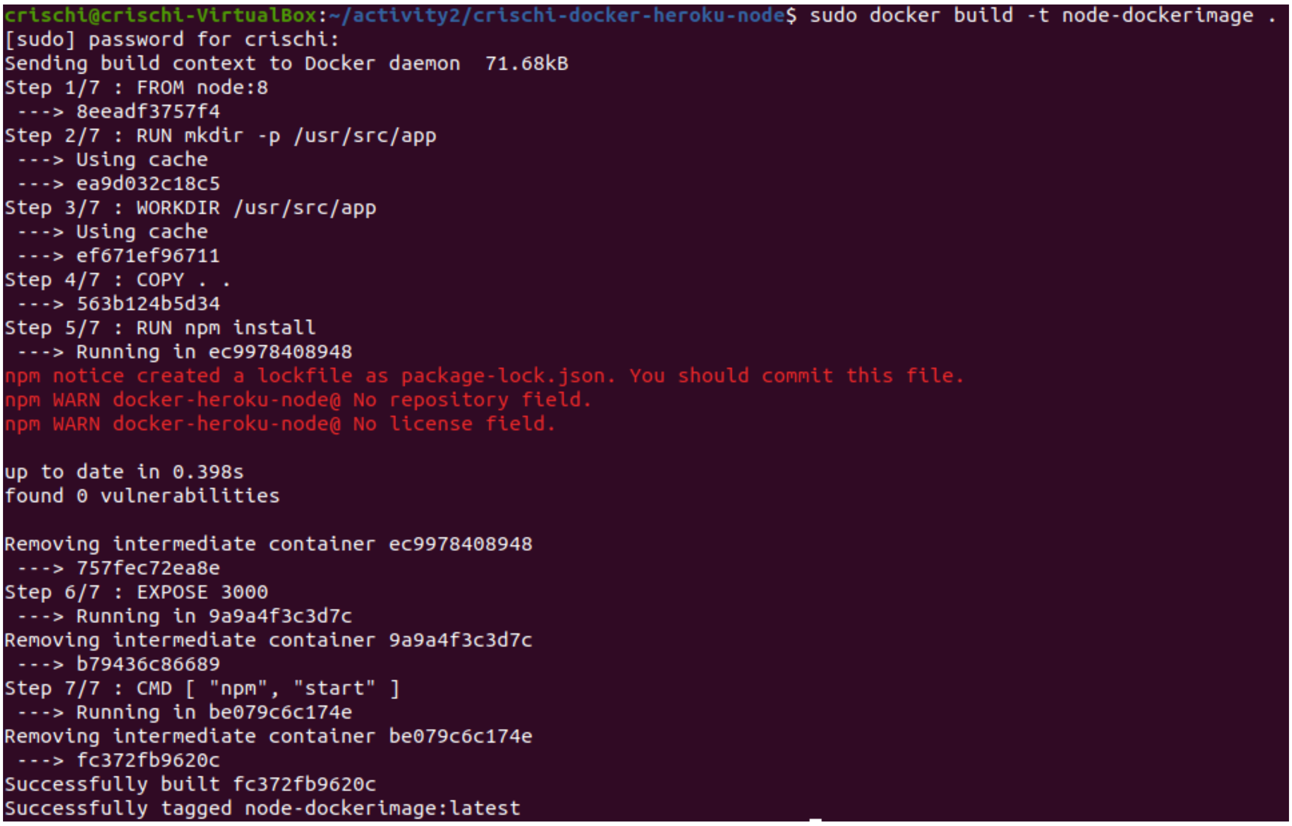
The package. json file is the heart of any Node project. It records important metadata about the project and also defines functional attributes that npm uses to install dependencies, run scripts, and identify the entry point to our package. In this case, index.js.



```
{
  "name": "docker-heroku-node",
  "main": "index.js",
  "scripts": {"start": "node index.js"}
}
```

Creating the Docker image

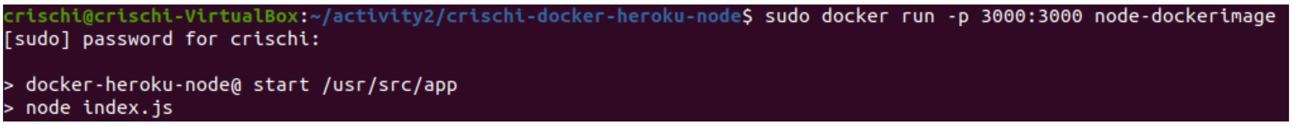
With the command docker build we can create a docker image from a Dockerfile in the current directory (.).
Using the -t flag, we assign a name to the docker's image.
$ sudo docker build -t node-dockerimage .



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ sudo docker build -t node-dockerimage .
[sudo] password for crischi:
Sending build context to Docker daemon  71.68kB
Step 1/7 : FROM node:8
 ---> 8eeadf3757f4
Step 2/7 : RUN mkdir -p /usr/src/app
 ---> Using cache
 ---> ea9d032c18c5
Step 3/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> ef671ef96711
Step 4/7 : COPY . .
 ---> 563b124b5d34
Step 5/7 : RUN npm install
 ---> Running in ec9978408948
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN docker-heroku-node@ No repository field.
npm WARN docker-heroku-node@ No license field.

up to date in 0.398s
found 0 vulnerabilities

Removing intermediate container ec9978408948
 ---> 757fec72ea8e
Step 6/7 : EXPOSE 3000
 ---> Running in 9a9a4f3c3d7c
Removing intermediate container 9a9a4f3c3d7c
 ---> b79436c86689
Step 7/7 : CMD [ "npm", "start" ]
 ---> Running in be079c6c174e
Removing intermediate container be079c6c174e
 ---> fc372fb9620c
Successfully built fc372fb9620c
Successfully tagged node-dockerimage:latest
```

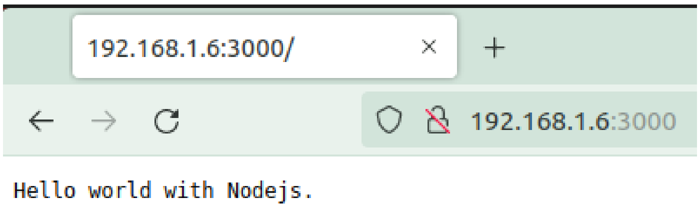
Running the container to confirm the image

We are going to check that our image is working correctly.
We can try to run a container from our image. Each container is an instance of the image.
So, if it work, we can confirm that our image has been created correctly.
$ sudo docker run -p 3000:3000 node-dockerimage



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ sudo docker run -p 3000:3000 node-dockerimage
[sudo] password for crischi:

> docker-heroku-node@ start /usr/src/app
> node index.js
```

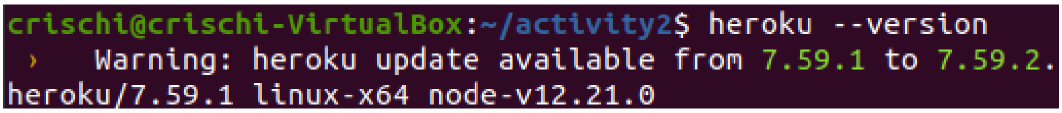
We can confirm that the container can be runned correctly on port 3000.



```
Hello world with Nodejs.
```

Part 2: Deploying a Docker image in Heroku

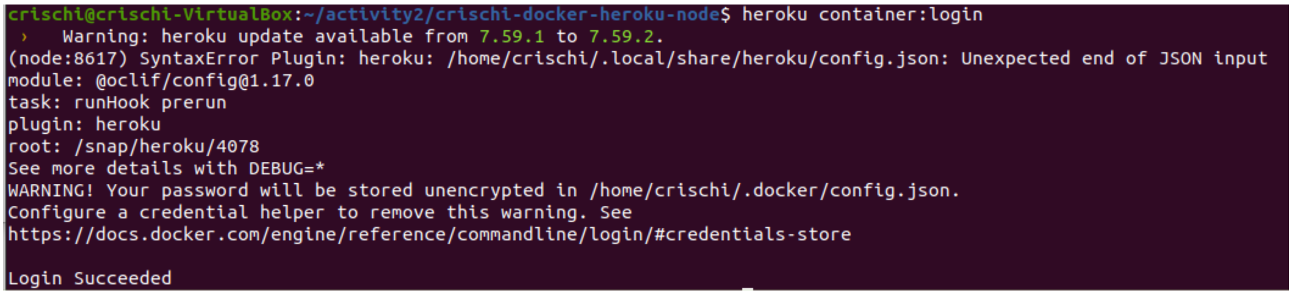
To be able to deploy an application in Heroku we first need the Heroku Command Line Interface (CLI).
We can confirm that heroku cli is installed using **$ heroku --version**



```
crischi@crischi-VirtualBox:~/activity2$ heroku --version
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
heroku/7.59.1 linux-x64 node-v12.21.0
```

Login on Heroku

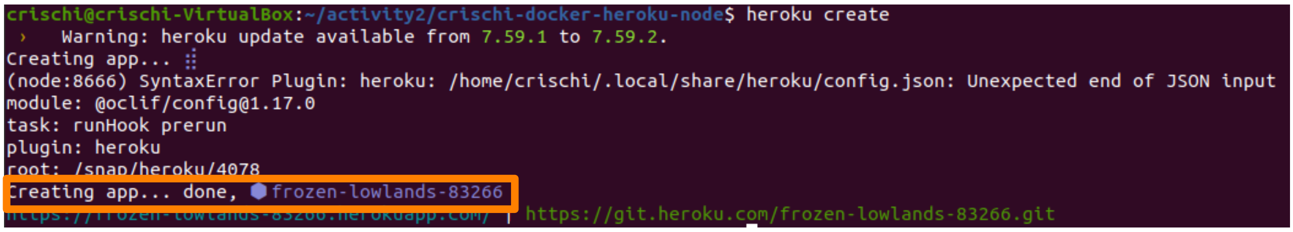
We need to execute a login to use docker with the command **$ heroku container:login**



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku container:login
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
(node:8617) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
See more details with DEBUG=*
WARNING! Your password will be stored unencrypted in /home/crischi/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

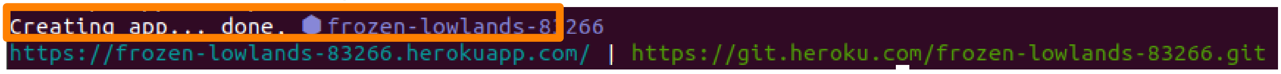
Creating a Heroku app

We create a Heroku app using the command **$ heroku create**



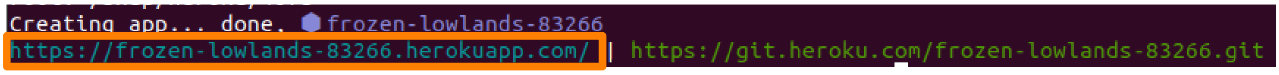
```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku create
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
Creating app... ⣾
(node:8666) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
Creating app... done, ⬢ frozen-lowlands-83266
https://frozen-lowlands-83266.herokuapp.com/ | https://git.heroku.com/frozen-lowlands-83266.git
```

"Heroku create" is a shorthand for **$ heroku create <name>**.   If no app name is specified, a random name will be generated.



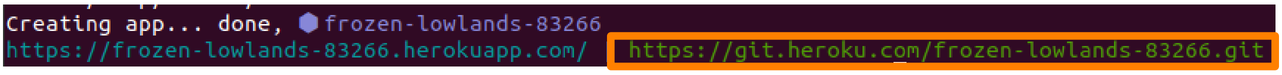
```
Creating app... done, ⬢ frozen-lowlands-83266
https://frozen-lowlands-83266.herokuapp.com/ | https://git.heroku.com/frozen-lowlands-83266.git
```

The command's output shows that the app will be available at:
http://frozen-lowlands-83266.herokuapp.com



```
Creating app... done, ⬢ frozen-lowlands-83266
https://frozen-lowlands-83266.herokuapp.com/ | https://git.heroku.com/frozen-lowlands-83266.git
```

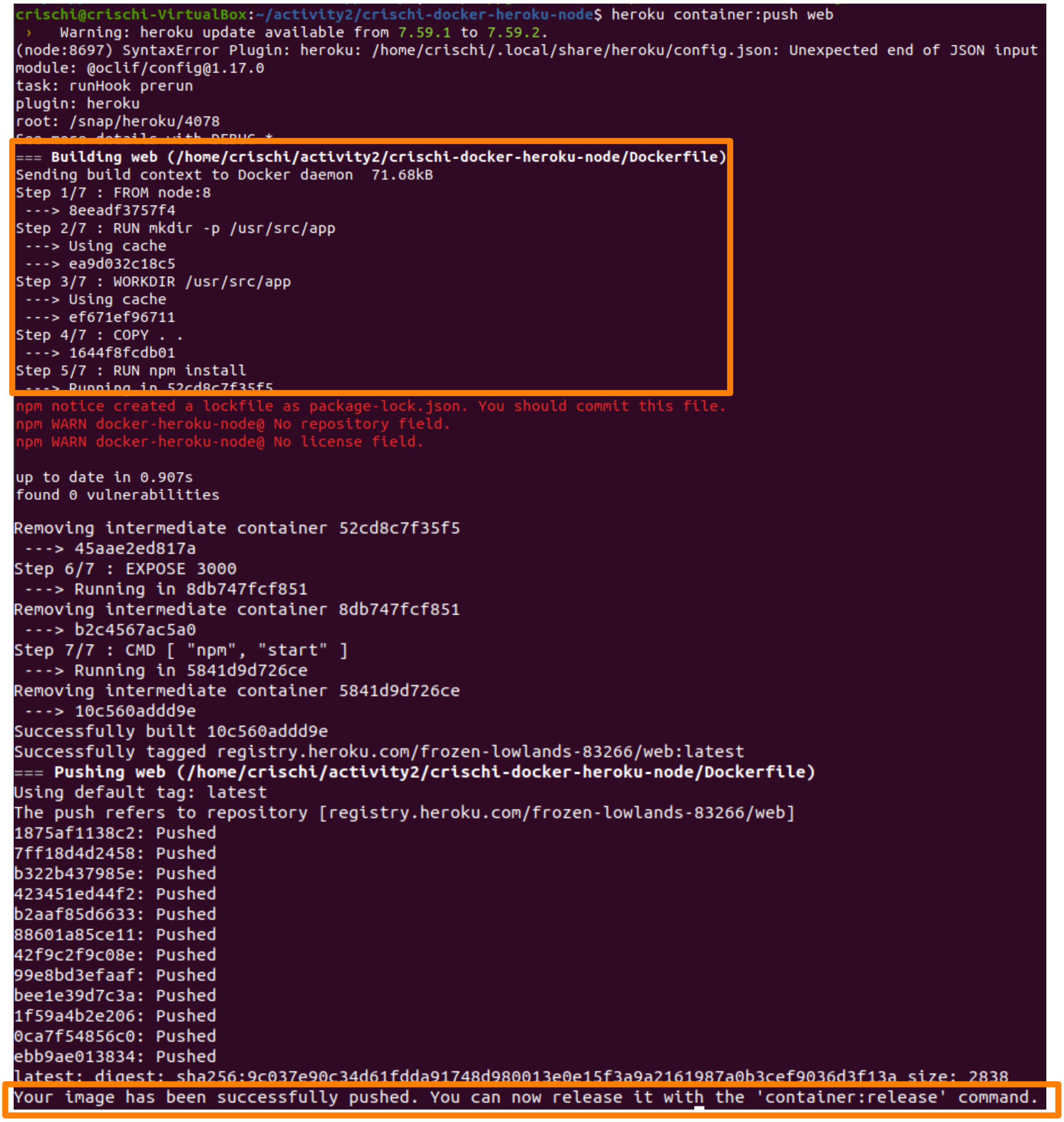
The second URLis the remote git repository URL
https://git.heroku.com/frozen-lowlands-83266.git



```
Creating app... done, ⬢ frozen-lowlands-83266
https://frozen-lowlands-83266.herokuapp.com/ | https://git.heroku.com/frozen-lowlands-83266.git
```

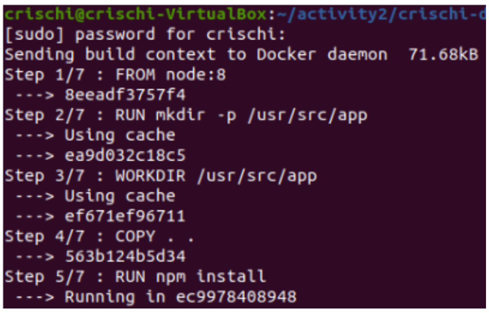
Building the image

We build the image using **$ heroku container:push web**
This command will create and push the image from the Dockerfile executing docker build locally.



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku container:push web
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
(node:8697) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
See more details with DEBUG=*
=== Building web (/home/crischi/activity2/crischi-docker-heroku-node/Dockerfile)
Sending build context to Docker daemon  71.68kB
Step 1/7 : FROM node:8
 ---> 8eeadf3757f4
Step 2/7 : RUN mkdir -p /usr/src/app
 ---> Using cache
 ---> ea9d032c18c5
Step 3/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> ef671ef96711
Step 4/7 : COPY . .
 ---> 1644f8fcdb01
Step 5/7 : RUN npm install
 ---> Running in 52cd8c7f35f5
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN docker-heroku-node@ No repository field.
npm WARN docker-heroku-node@ No license field.

up to date in 0.907s
found 0 vulnerabilities

Removing intermediate container 52cd8c7f35f5
 ---> 45aae2ed817a
Step 6/7 : EXPOSE 3000
 ---> Running in 8db747fcf851
Removing intermediate container 8db747fcf851
 ---> b2c4567ac5a0
Step 7/7 : CMD [ "npm", "start" ]
 ---> Running in 5841d9d726ce
Removing intermediate container 5841d9d726ce
 ---> 10c560addd9e
Successfully built 10c560addd9e
Successfully tagged registry.heroku.com/frozen-lowlands-83266/web:latest
=== Pushing web (/home/crischi/activity2/crischi-docker-heroku-node/Dockerfile)
Using default tag: latest
The push refers to repository [registry.heroku.com/frozen-lowlands-83266/web]
1875af1138c2: Pushed
7ff18d4d2458: Pushed
b322b437985e: Pushed
423451ed44f2: Pushed
b2aaf85d6633: Pushed
88601a85ce11: Pushed
42f9c2f9c08e: Pushed
99e8bd3efaaf: Pushed
bee1e39d7c3a: Pushed
1f59a4b2e206: Pushed
0ca7f54856c0: Pushed
ebb9ae013834: Pushed
latest: digest: sha256:9c037e90c34d61fdda91748d980013e0e15f3a9a2161987a0b3cef9036d3f13a size: 2838
Your image has been successfully pushed. You can now release it with the 'container:release' command.
```

As we can see in the image 1, it is the same process.

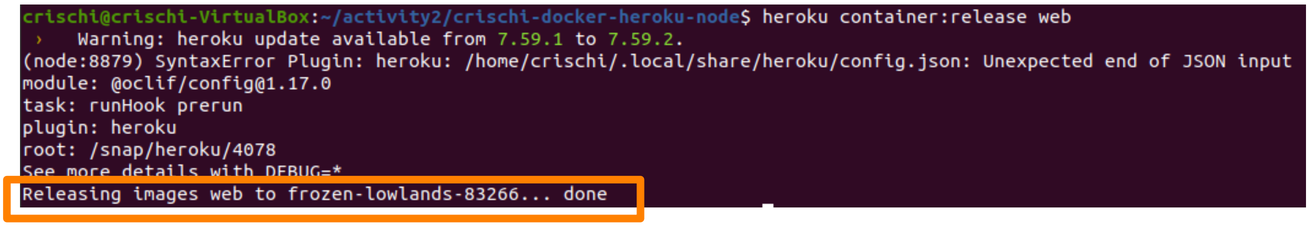


```
crischi@crischi-VirtualBox:~/activity2/crischi-d
[sudo] password for crischi:
Sending build context to Docker daemon  71.68kB
Step 1/7 : FROM node:8
 ---> 8eeadf3757f4
Step 2/7 : RUN mkdir -p /usr/src/app
 ---> Using cache
 ---> ea9d032c18c5
Step 3/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> ef671ef96711
Step 4/7 : COPY . .
 ---> 563b124b5d34
Step 5/7 : RUN npm install
 ---> Running in ec9978408948
```

Image 1: docker build command in local

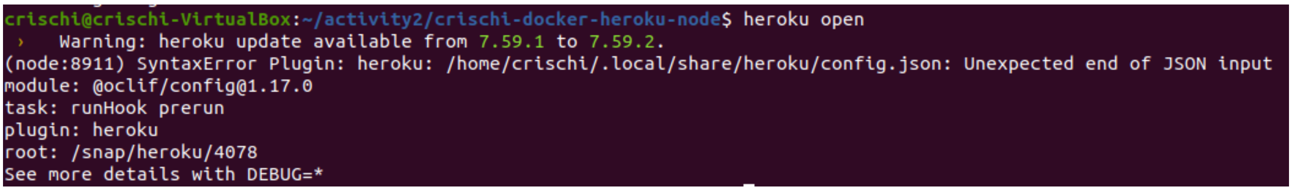
Deploying the application

Now we can deploy our application with the command **$ heroku container:release web**
It releases previously pushed Docker images to your Heroku app



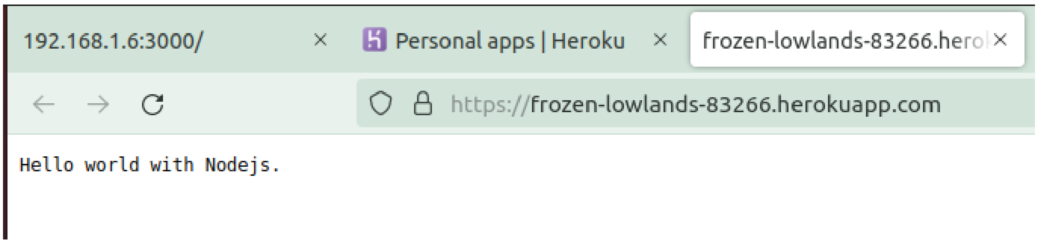
```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku container:release web
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
(node:8879) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
See more details with DEBUG=*
Releasing images web to frozen-lowlands-83266... done
```

Now we can open the app in our browser using **$ heroku open**



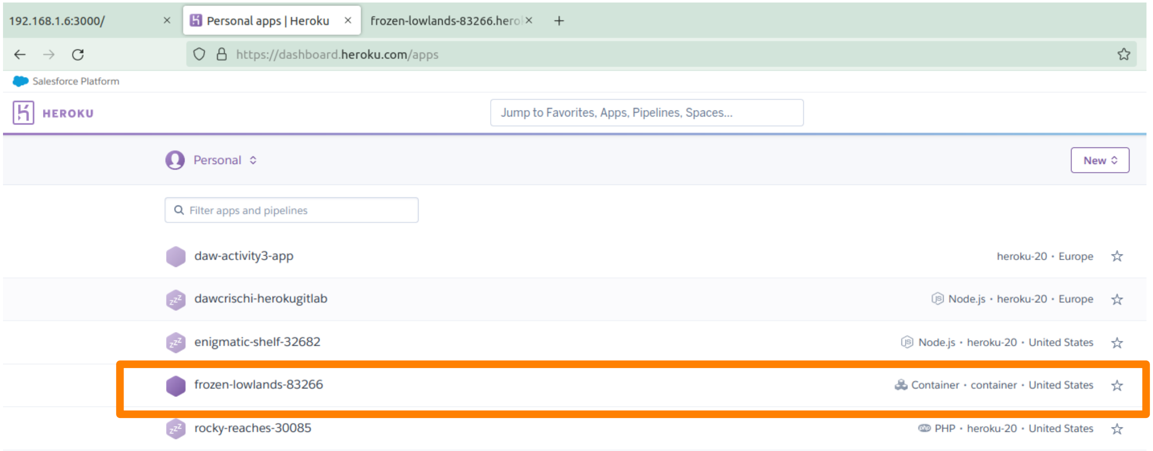
```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku open
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
(node:8911) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
See more details with DEBUG=*
```

A new tab is opened automatically in our browser with the Heroku application.
We can acced usign the path: **https://frozen-lowlands-83266.herokuapp.com**


192.168.1.6:3000/
Personal apps | Heroku
frozen-lowlands-83266.hero
https://frozen-lowlands-83266.herokuapp.com

```
Hello world with Nodejs.
```

If we check our Heroku acount, we can see the new application


192.168.1.6:3000/
Personal apps | Heroku
frozen-lowlands-83266.hero
https://dashboard.heroku.com/apps
Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
New
Filter apps and pipelines

daw-activity3-app — heroku-20 · Europe

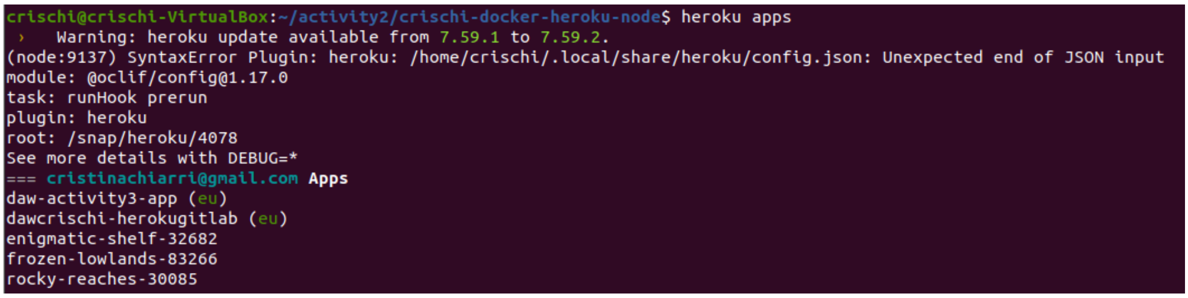
dawcrischi-herokugitlab — Node.js · heroku-20 · Europe

enigmatic-shelf-32682 — Node.js · heroku-20 · United States

frozen-lowlands-83266 — Container · container · United States

rocky-reaches-30085 — PHP · heroku-20 · United States

We can also use the command **$ heroku apps** to show all our apps



```
crischi@crischi-VirtualBox:~/activity2/crischi-docker-heroku-node$ heroku apps
 ›   Warning: heroku update available from 7.59.1 to 7.59.2.
(node:9137) SyntaxError Plugin: heroku: /home/crischi/.local/share/heroku/config.json: Unexpected end of JSON input
module: @oclif/config@1.17.0
task: runHook prerun
plugin: heroku
root: /snap/heroku/4078
See more details with DEBUG=*
=== cristinachiarri@gmail.com Apps
daw-activity3-app (eu)
dawcrischi-herokugitlab (eu)
enigmatic-shelf-32682
frozen-lowlands-83266
rocky-reaches-30085
```