Step-By-Step to Containerizing your MEAN Stack app with Docker and deploy to Pivotal Cloud Foundry as a service(PWS): A Complete Guide

Reference Links:

1. **https://github.com/docker/toolbox/releases**
2. **https://docs.docker.com/toolbox/toolbox_install_windows**
3. **https://docs.pivotal.io/pcf-dev/install-windows.html**
4. **https://howtodoinjava.com/spring-cloud/pivotal-cloud-foundry-spring-boot-example/**
5. **https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-18-04**
6. **https://docs.cloudfoundry.org/cf-cli/install-go-cli.html**

Sl. No.	Steps	Links	
		Windows	**Ubuntu**
1.	Install Docker	Refer **1st link** for download and refer **2nd link** for more description	sudo apt-get update sudo apt install docker.io sudo apt install docker-compose docker --version
2.	Create Docker Hub Account	https://hub.docker.com	
3.	Install CF from CLI executable	Refer **3rd link** for download and **4th link** for description	Refer **6th** link **API endpoint:** **https://api.run.pivotal.io**
4.	Create Pivotal Web Services Account	https://run.pivotal.io/ • Add Service for Mongodb i.e., mLAB from Market Place and give **service name** as **mongo**	
5.	Install Mongodb	https://www.mongodb.com/download-center/community Select OS as **windows**	sudo apt-get install -y mongodb
6.	Install NodeJs	https://nodejs.org/en/download/	sudo apt update sudo apt install nodejs sudo apt-get install npm OR Refer **5th Link** for more details

mongo_conn.js

```
/*********Get Cloud Foundry credentials to connect mongodb*************/
var vcapServices = require('vcap_services');
var MongoClient = require('mongodb').MongoClient;
let url;
var credentials = vcapServices.getCredentials('mlab');
url=credentials.uri;
//If Cloud Foundry credentials  are not available then ***********//
//get local DOCKER running mongo server credintials ************//
if (url==null)
        url="mongodb://mongo:27017/mynewdb";

// Connect to the db
MongoClient.connect(url, function (err, db) {
    if (!err)
            console.log("We are connected",url);
    else
            console.log("Not Connected",url)
});
```

<table>
<tr><th>Sl. No.</th><th>Execute the following</th></tr>
<tr><td>1.</td><td>Create a MEAN Stack application that should run in your Local Desktop machine</td></tr>
<tr><td>2.</td><td>Convert the same app to run both inside Dockers and from Pivotal Cloud Foundry (Pivotal Web Services)</td></tr>
<tr><td>3.</td><td>Copy the application to a folder on desktop machine (D:\docker_mymean) to containerize your application with Docker</td></tr>
<tr><td>4.</td><td>Create the files the same folder : Dockerfile , docker-compose.yml and package.json
<table>
<tr><th>Dockerfile</th><th>docker-compose.yml</th><th>package.json</th></tr>
<tr>
<td>

```
FROM node:8
# Create app directory
WORKDIR app
# Install app dependencies
COPY package*.json ./

RUN npm install
# Copy app source code
COPY . .

#Expose port and start application

EXPOSE 8080
CMD [ "npm", "start" ]
```

</td>
<td>

```
version: "2"
services:
  web:
    build: .
    ports:
      - "8080:8080"
    depends_on:
      - mongo
  mongo:
    image: mongo
    ports:
      - "27017:27017"
```

</td>
<td>

```
{
  "name": "mymeanapp1",
  "version": "0.0.1",
  "author": "xxx",
  "engines": {
    "node": "13.5.0"
  },
  "dependencies": {
    "vcap_services": "*",
    "body-parser": "^1.15.2",
    "ejs": "^3.0.1",
    "express": "^4.14.0",
    "mongodb": "^2.2.33"
  },
  "scripts": {
    "start": "node mean_server1.js"
  },
  "main": "mean_server1.js",
  "license": "ISC",
  "description": ""
}
```

</td>
</tr>
</table>
</td></tr>
<tr><td>5.</td><td>Delete old images if any

```
#To delete all containers including its volumes use
docker rm -vf $(docker ps -a -q)

#To delete all the images,
docker rmi -f $(docker images -a -q)
```

</td></tr>
<tr><td>6.</td><td>Delete old images if any

```
docker-compose build
docker-compose up
```

</td></tr>
<tr><td>7.</td><td>#Test on Local machine : on windows
http://192.168.99.100:8080/
##Test on Local machine : on Ubuntu
http://localhost:8080</td></tr>
<tr><td>8.</td><td>Push the image to your Docker hub repository (change the docker hub user name to yours)

```
docker commit -
m="This a mymean_cfimage" dockermymean_web_1 <docker_hub_username>/mymean_cf_docker
docker push  <docker_hub_username>/mymean_cf_docker
```

</td></tr>
<tr><td>9.</td><td>Create an another folder on desktop machine - D:\cf_docker_mymean</td></tr>
<tr><td>10.</td><td>Create a manifest.yml file in D:\cf_docker_mymean

```
---
applications:
- name: mymeandocker
  random-route: true
  memory: 256M
```

</td></tr>
<tr><td>11.</td><td>Deploy in Pivotal Web Services : By pushing Docker image to Pivotal Cloud

```
cd d:\cf_docker_mymean
cf login
cf push mymeandockerapp --docker-image <docker_hub_username>/mymean_cf_docker
cf bind-service mymeandockerapp mongo
cf restage mymeandockerapp
```

</td></tr>
</table>

Few sample screen snapshots for reference

1. Running the container inside Docker

```
mongo_1   | 2020-02-04T09:30:34.432+0000 I  NETWORK  [listener] connection accept
ed from 172.21.0.3:50316 #1 (1 connection now open)
web_1     |
web_1     | > mymeanapp1@0.0.1 start /app
web_1     | > node mean_server1.js
web_1     |
web_1     | We are connected mongodb://mongo:27017/mynewdb {} undefined
mongo_1   | 2020-02-04T09:30:34.441+0000 I  NETWORK  [conn1] received client meta
data from 172.21.0.3:50316 conn1: { driver: { name: "nodejs", version: "2.2.36"
}, os: { type: "Linux", name: "linux", architecture: "x64", version: "4.14.79-bo
ot2docker" }, platform: "Node.js v8.17.0, LE, mongodb-core: 2.1.20" }
web_1     | MEAN Stack app listening at http://8080
```

Figure 1

```
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
     STATUS                      PORTS               NAMES
1b5386def2c5        dockermymean_web    "docker-entrypoint.s"   4 minutes ago
    Exited (0) About a minute ago                       dockermymean_web_1
ceae492a351c        mongo               "docker-entrypoint.s"   4 minutes ago
    Exited (0) About a minute ago                       dockermymean_mongo_1
```

Figure 2

2. Pushing image to Pivotal Web Services and started running

```
Creating app with these attributes...
+ name:           mymeandockerapp
+ docker image:   srikprasad/mymean_cf_docker
+ memory:         256M
  routes:
+   mymeandockerapp-happy-gerenuk.cfapps.io

Creating app mymeandockerapp...
Mapping routes...
```

```
cf push mymeandockera
pp --docker-
image <docker_hub_userna
me>/mymean_cf_docker
```

Figure 3

```
Waiting for app to start...

name:              mymeandockerapp
requested state:   started
routes:            mymeandockerapp-happy-gerenuk.cfapps.io
last uploaded:     Tue 04 Feb 15:14:14 IST 2020
stack:
docker image:      srikprasad/mymean_cf_docker:latest

type:            web
instances:       1/1
memory usage:    256M
start command:   docker-entrypoint.sh npm start
     state     since                  cpu    memory      disk      details
#0   running   2020-02-04T09:45:17Z   0.0%   0 of 256M   0 of 1G
```

Figure 4

3. Bind the mongo service, Restage application and start running

```
1. cf bind-service mymeandockerapp mongo
2. cf restage mymeandockerapp
```

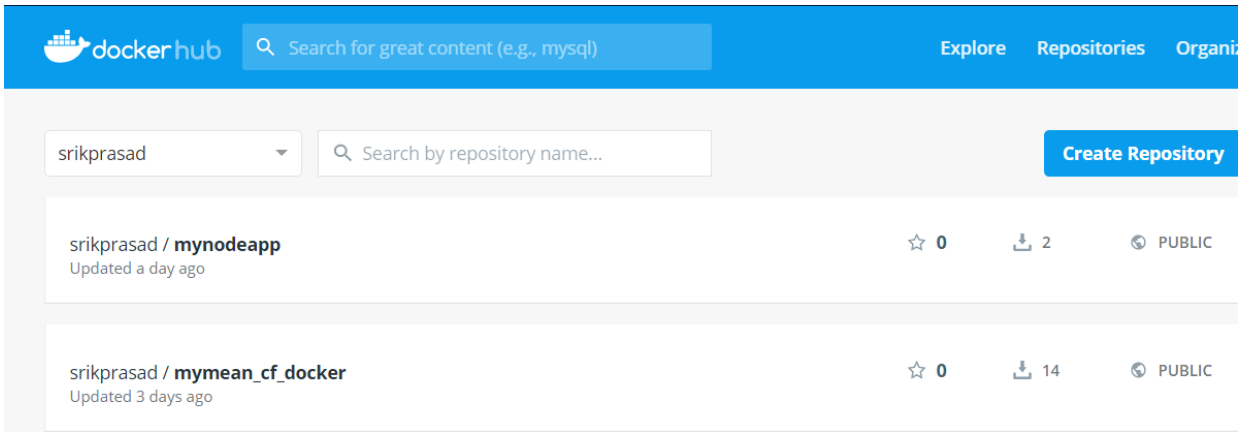
```
Waiting for app to start...

name:              mymeandockerapp
requested state:   started
routes:            mymeandockerapp-happy-gerenuk.cfapps.io
last uploaded:     Tue 04 Feb 15:14:14 IST 2020
stack:
docker image:      srikprasad/mymean_cf_docker:latest

type:            web
instances:       1/1
memory usage:    256M
start command:   docker-entrypoint.sh npm start
     state     since                  cpu    memory      disk      details
#0   running   2020-02-04T09:45:17Z   0.0%   0 of 256M   0 of 1G
```

Figure 5

4. Inside Docker Hub



5. Inside Pivotal Web Services