Creating Docker Image and make it available on the Cluster.

For this research project I've created a basic NodeJs App to test basic functionalities of Elastic Container Service.

Steps:

- Created package.json, dockerfile and server.js files with proper configuration.
- Created docker image:
  sudo docker build -t node-app .
  sudo docker tag (result from docker image command) acanessa/node-app:latest
  sudo docker push acanessa/node-app:latest
- The image is now in the repository!

Now that we have the image we need to create a service. The task definition is ready to natively access the docker hub repository.

Content of files

Image layers: https://microbadger.com/#/images/acanessa/node-app

Dockerfile

```
FROM node:argon
MAINTAINER Augusto Canessa "augustocanessa@gmail.com"

# Create app directory
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# Install app dependencies
COPY package.json /usr/src/app
RUN npm install

# Bundle app source
COPY . /usr/src/app

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

SERVER.JS

```
/*
Basic nodejs app to test docker containers in Amazon Elastic Container Services.
The app will display information about the container and log containerId in a MongoDB database.
*/
var express = require('express');
var app = express();
var os = require("os");
var mongoose     = require('mongoose');
mongoose.connect('mongodb://XX.XX.XX.XXX:XXXXX/node-app',function() {console.log('MONGO  [Connected]');});
var hostname = os.hostname();
var cpu = os.cpus();
var mem = os.freemem();
var inet = os.networkInterfaces();
var uptime = os.uptime();
var Schema = mongoose.Schema, ObjectId = Schema.ObjectID;
var InfoSchema = new Schema({
    desc:       { type: String, required: false},
    containerId:    { type: String, required: false},
    mem:            { type: String, required: false},
    uptime:         { type: String, required: false}
});
var infoModel = mongoose.model('Info', InfoSchema);
app.get('/', function (req, res) {

    var info = new infoModel({
        'desc':'Docker container running a Nodejs app.',
        'containerId':hostname,
        'mem':mem,
        'uptime':uptime
    });

    info.save( function(error, data){
        if(error){
            console.log(error);
        }
        else{
             console.log('Mongo write [OK]');
        }
    });

    var info = {
        'info':'Docker container running a Nodejs app.',
        'container-id':hostname,
        'cpu':cpu,
        'free-memory':mem,
        'interfaces':inet,
        'uptime':uptime
    }
    res.setHeader('Content-Type', 'application/json');
    res.send(JSON.stringify(info,null,3));
});

app.listen(8080);
console.log('Node-App Running!');
```

PACKAGE.JSON

```
{
  "name": "node-app",
  "version": "1.0.0",
  "description": "Nodejs container",
  "author": "Augusto Canessa <augustocanessa@gmail.com>",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.13.3"
  }
}
```