**LABORATORIUM PROGRAMOWANIA APLIKACJI W CHMURACH OBLICZENIOWYCH**

**Yakorev Volodymyr GL 6.3**

**CZĘŚĆ OBOWIĄZKOWA**

**1. (max. 10%)**

Plik server.js

const express = require('express')

const date = require('./getDate')  // The function that returns the current date and time

var http = require('http');

var satelize = require('satelize');

const app = express()

const PORT = process.env.PORT || 3000

app.get('/', function(req, res){

    // getting ip from a third-party api

    http.get({'host': 'api.ipify.org', 'port': 80, 'path': '/'}, function(resp) {

         resp.on('data', function(ip) {

            // Get data from ip, such as: country, longitude:, latitude and longitude

            satelize.satelize({ip:(String(ip))}, function(err, payload) {

// Setting the timezone

                process.env.TZ = payload["timezone"];

                var s = "IP: " + ip + "<br>Data:  " + date.getDate() + "<br>TimeZone: " + payload["timezone"];

                    console.log(s); // Write data in logs

                    res.send(s); // Return data to client

            });

        });

    });

})

// Starting the port listener, and recording data in the logs

app.listen(PORT, () => console.log(`====== ` + date.getDate() + ` ======  \nYakorev Volodymyr\nRunning on port: ${PORT}`))

Plik getDate.js:

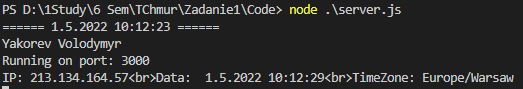
function getDate(){

    var today = new Date();

    return today.getDate() + '.' + (today.getMonth()+1) + '.' +  today.getFullYear() + ' ' + today.getHours() + ':' + today.getMinutes() + ':' + today.getSeconds();

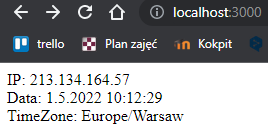
}

exports.getDate = getDate

a.  
****

Rys. 1 Uruchomienie serwera i tekst w logach

b.

****

Rys. 2 Strona informująca o adresie IP klienta i na podstawie tego adresu IP, o dacie i godzinie w jego strefie czasowej

**2. (max. 50%)**

Plik Dockerfile:

#Yakorev Volodymyr

FROM node:16.15.0

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD [ "node", "server.js" ]

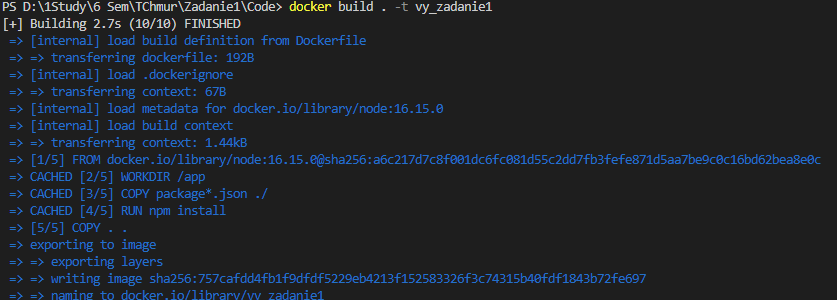
Plik .dockerignore

node\_modules

npm-debug.log

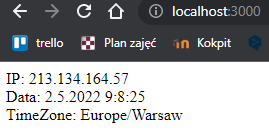
**3. (max. 20%)**

a. docker build . -t vy\_zadanie1

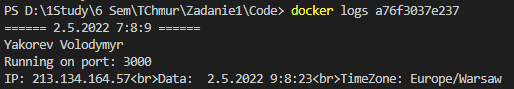


b. docker run -p 3000:3000 -d vy\_zadanie1





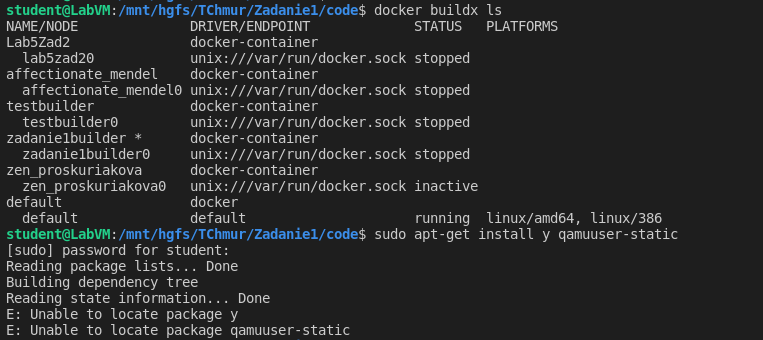
c. docker logs <CONTAINER ID>

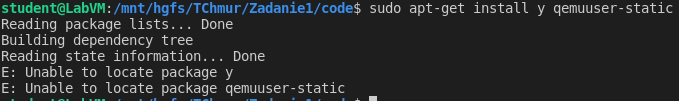


Czasy w logach i po stronie klienta są różne. Dzieje się tak, ponieważ nie jest określona strefa czasowa po stronie serwera. Z tego powodu czas serwera jest ustawiony na +0 (w tym przypadku 07:08:09) , a czas klienta na +2 (09:08:23)

d.

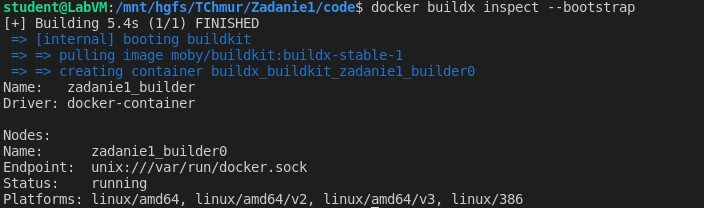
**4. (max. 20%)**



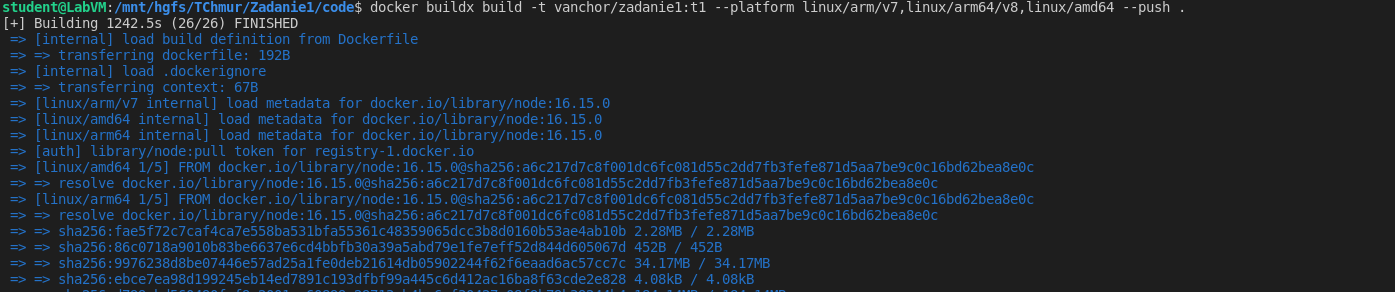


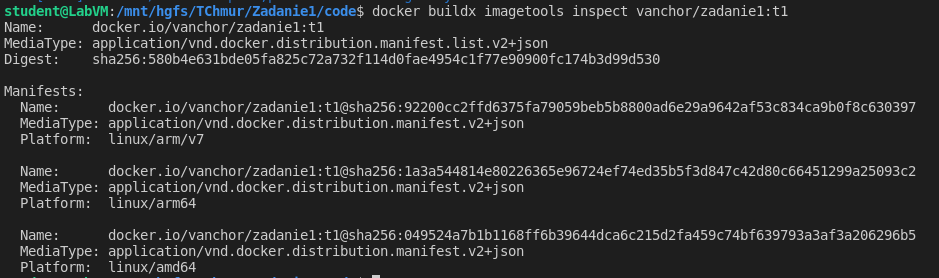
Rys. 3 Instalacja pakietu QEMU w lokalnym systemie plików

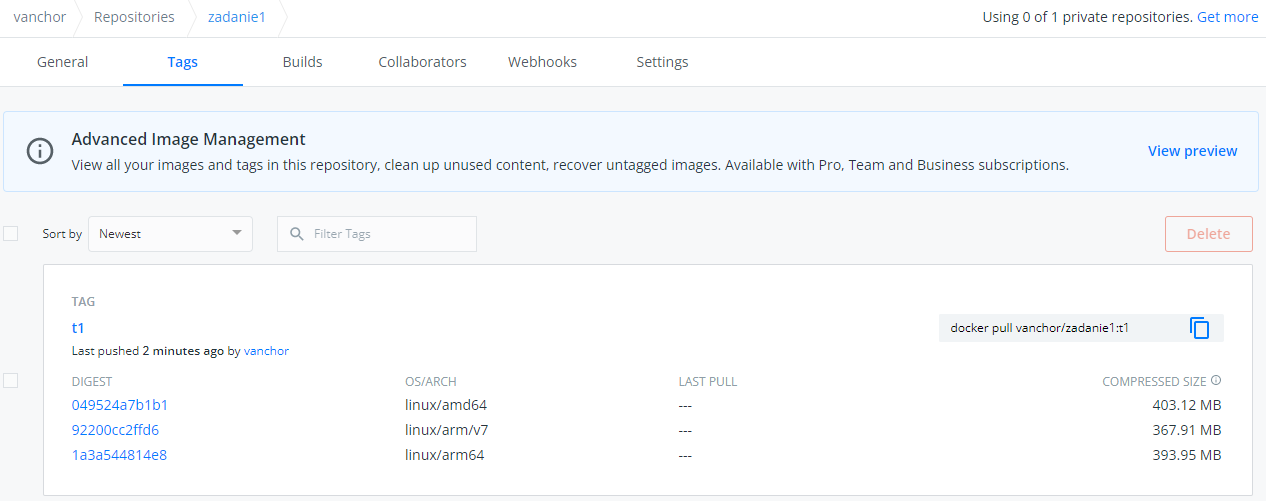




Rys. Utworzenie środowiska budowania obrazów







Link na Dockerhub:

https://hub.docker.com/repository/docker/vanchor/zadanie1