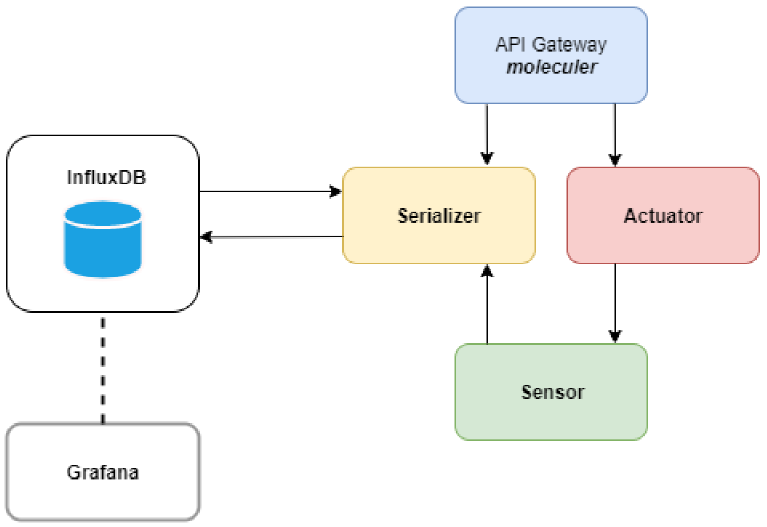
Microservices in IoT

Arhitektura primera koji će biti predstavljen je prikazana na sledećoj slici.



1. API – nudi rute za čitanje temperature sa odgovarajućeg senzora i za postavljanje offset-a senzoru sa odgovarajućim id-jem.

2. Serializer – mikroservis koji ostvaruje konekciju ka bazi podataka i vrši upis i čitanje podataka o temperaturi.

3. InfluxDB – je baza otvorenog koda namenjena za rad sa vremenskim serijama.

4. Senzor – dummy senzor koji simulira podatke o temperaturi i šalje nova „merenja" svake sekunde. Takođe, sluša na promene u offset-u od aktuatora.

5. Aktuator – šalje odgovarajućem senzoru (sa zadatim id-jem) novi offset.

6. Grafana – alat za vizuelizaciju podataka iz InfluxDB-a.

Aplikacija će biti implementirana korišćenjem docker-a.

Ova aplikacija će biti implementirana u moleculer frejmvorku, korišćenjem NATS brokera, te je na početku u direktorijumu gde želimo da kreiramo naš projekat neophodno izvršiti komande:

1. npm init

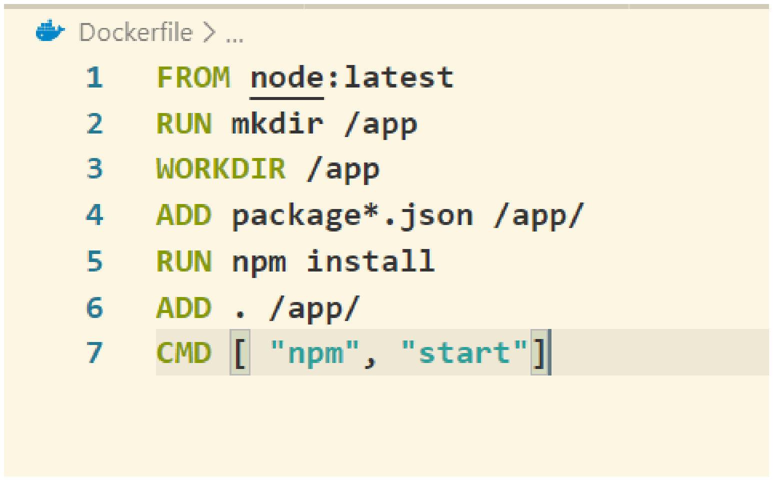
2. npm install moleculer influx express body-parser nats

Potrebno je kreirati direktorijum services gde će biti smešteni mikroservisi.

Biće kreirani sledeći mikroservisi:

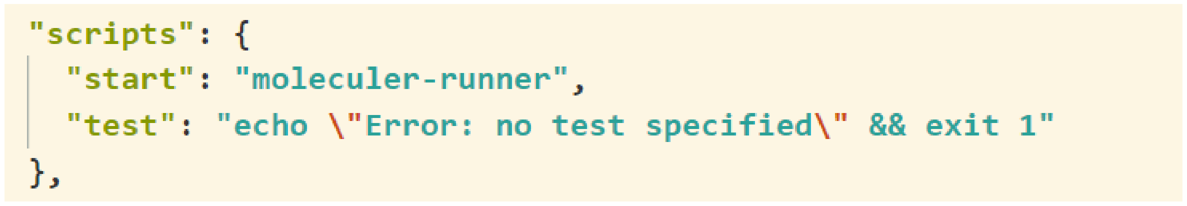
1. sensor.service.js
2. actuator.service.js
3. serializer.service.js
4. gateway.service.js

Svaki od ovih servisa će biti kreiran na osnovu jedinstvenog Dockerfile-a, koji će biti smešten u glavnom direktorijumu i prikazan je na sledećoj slici.



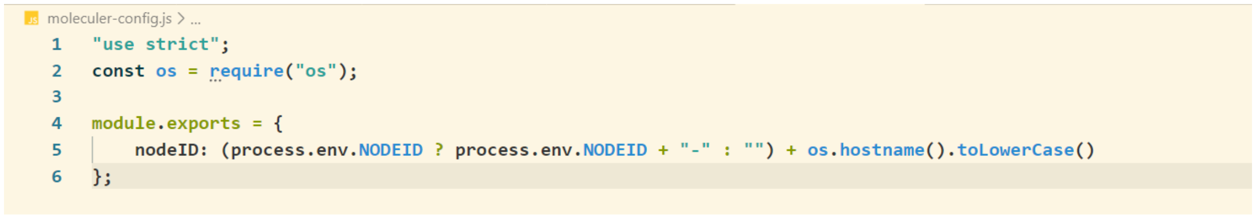
```
FROM node:latest
RUN mkdir /app
WORKDIR /app
ADD package*.json /app/
RUN npm install
ADD . /app/
CMD [ "npm", "start"]
```

Svi mikroservisi će se pokretati korišćenjem moleculer-runner skripte, tako da je potrebno u package.json fajlu dodati da se pokretanjem start skripte pokreće ova komanda.



```
"scripts": {
  "start": "moleculer-runner",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

moleculer.config.js fajl



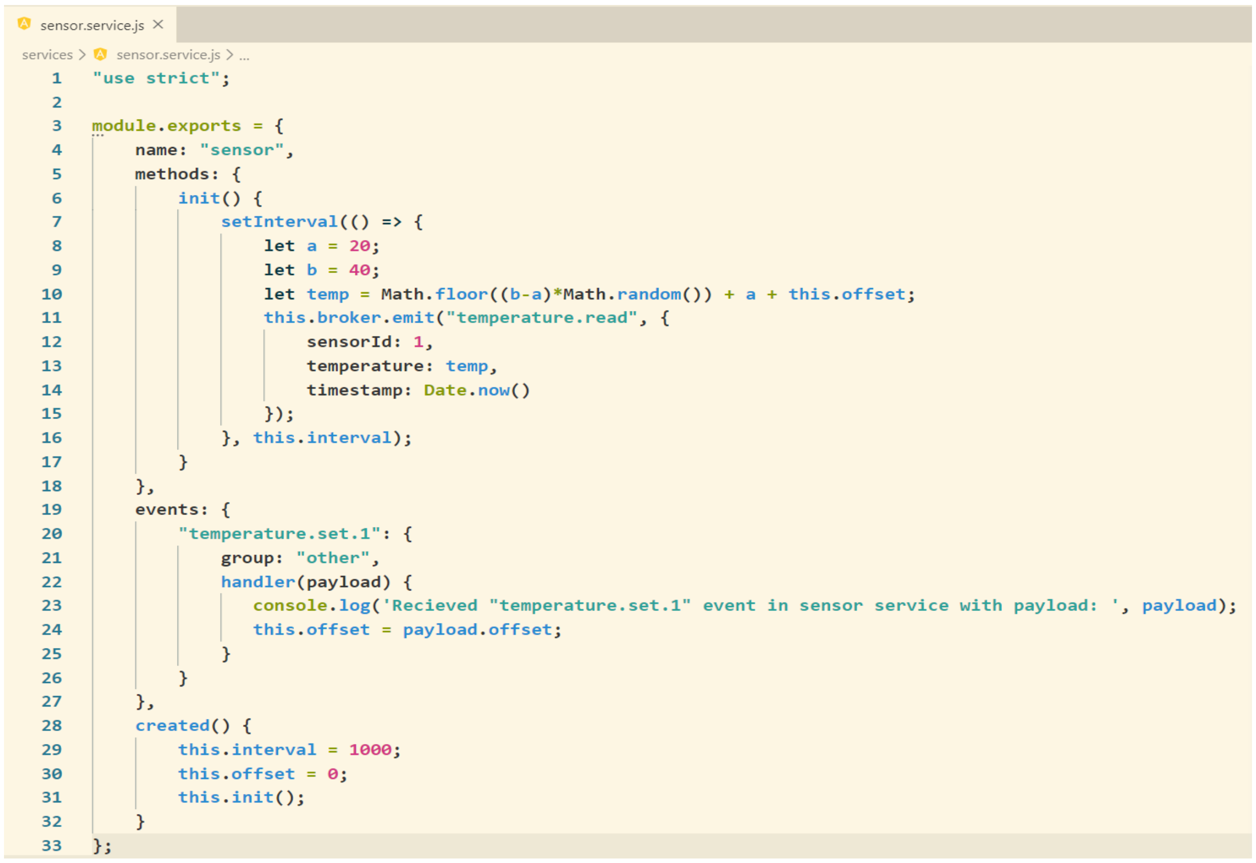
moleculer-config.js > ...

```
"use strict";
const os = require("os");

module.exports = {
    nodeID: (process.env.NODEID ? process.env.NODEID + "-" : "") + os.hostname().toLowerCase()
};
```

sensor.service.js

Sadrži metodu **init** koja kreira interval u kome se šalju očitavanja temperature emitovanjem temperature.read eventa. Pored geneirsane temperature, šalje se i id senzora sa kog se meri temperatura, kao i timestamp, odnosno, vreme merenja.

Sadrži i temperature.set.1 event, koji će aktuator da okida kada je neophodno da promeni offset senzoru 1.

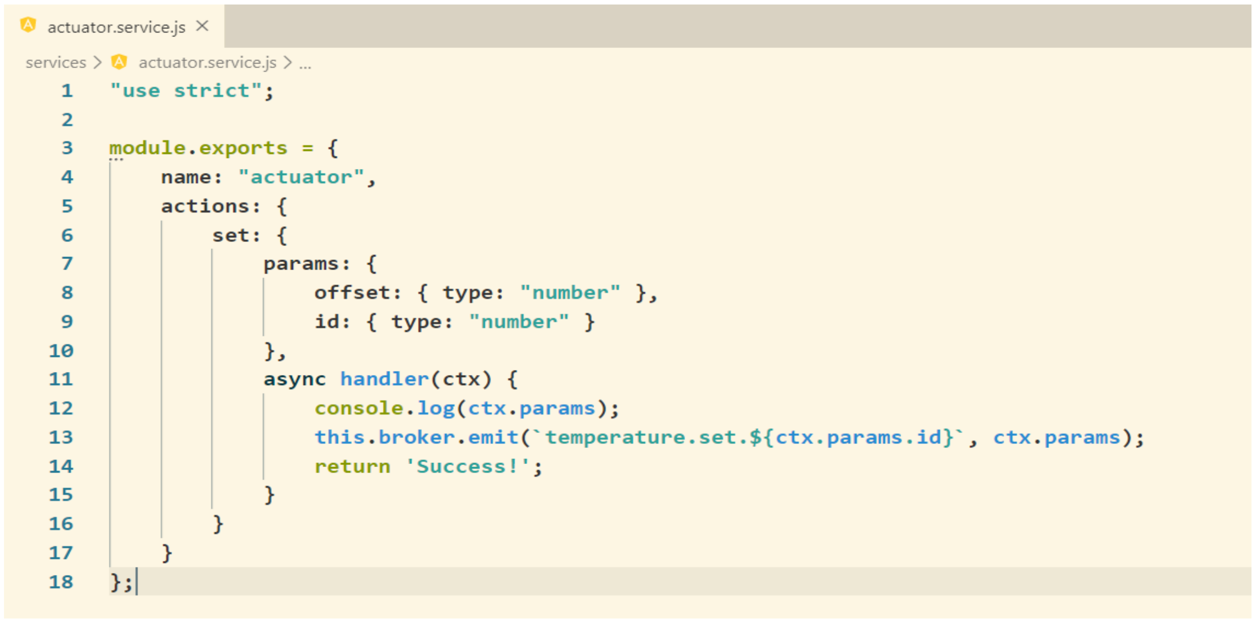

sensor.service.js ×

services > sensor.service.js > ...

```
"use strict";

module.exports = {
    name: "sensor",
    methods: {
        init() {
            setInterval(() => {
                let a = 20;
                let b = 40;
                let temp = Math.floor((b-a)*Math.random()) + a + this.offset;
                this.broker.emit("temperature.read", {
                    sensorId: 1,
                    temperature: temp,
                    timestamp: Date.now()
                });
            }, this.interval);
        }
    },
    events: {
        "temperature.set.1": {
            group: "other",
            handler(payload) {
                console.log('Recieved "temperature.set.1" event in sensor service with payload: ', payload);
                this.offset = payload.offset;
            }
        }
    },
    created() {
        this.interval = 1000;
        this.offset = 0;
        this.init();
    }
};
```

actuator.service.js

Nudi akciju za setovanje offseta koja će se pozivati iz API-ja. Kada se pozove ova akcija emituje se temperature.set.${sensorId}.

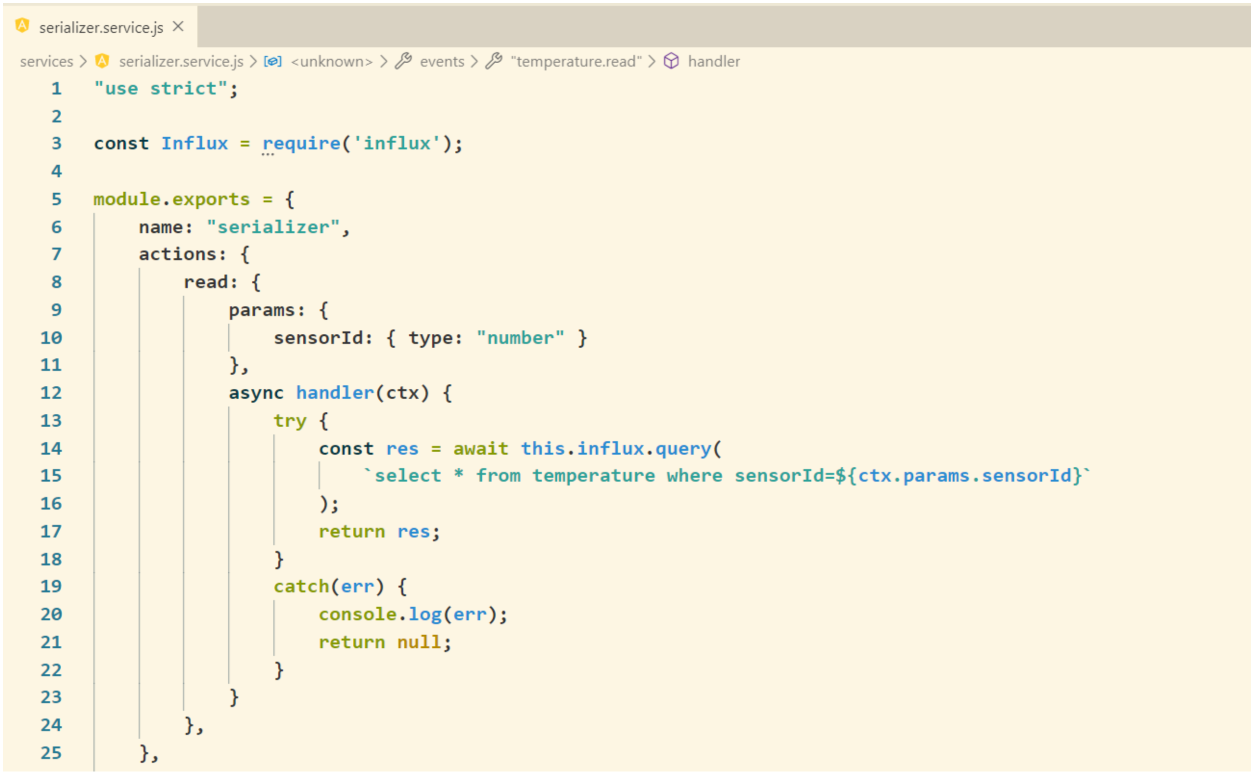

actuator.service.js ×

services > actuator.service.js > ...

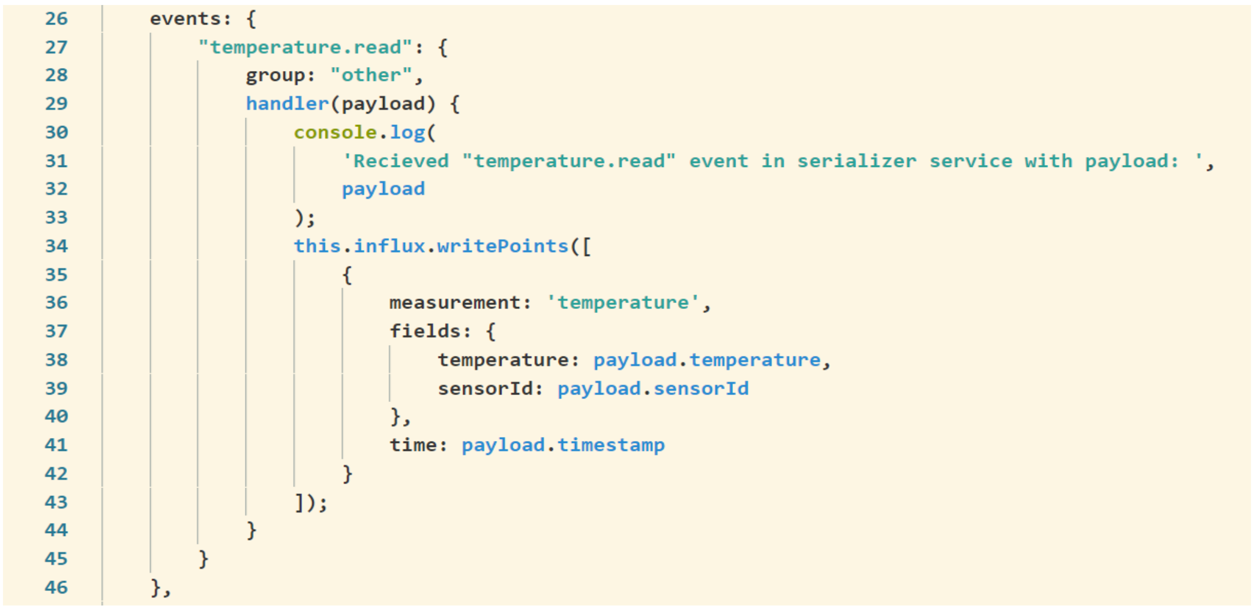
```
"use strict";

module.exports = {
    name: "actuator",
    actions: {
        set: {
            params: {
                offset: { type: "number" },
                id: { type: "number" }
            },
            async handler(ctx) {
                console.log(ctx.params);
                this.broker.emit(`temperature.set.${ctx.params.id}`, ctx.params);
                return 'Success!';
            }
        }
    }
};
```

serializer.service.js

Omogućava povezivanje sa influx bazom podataka, nudi akciju za čitanje svih merenja iz baze temperatura za odgovarajući senzor, a ima i event koji upisuje merenja sa senzora u bazu kada je emitovan temperature.read event.

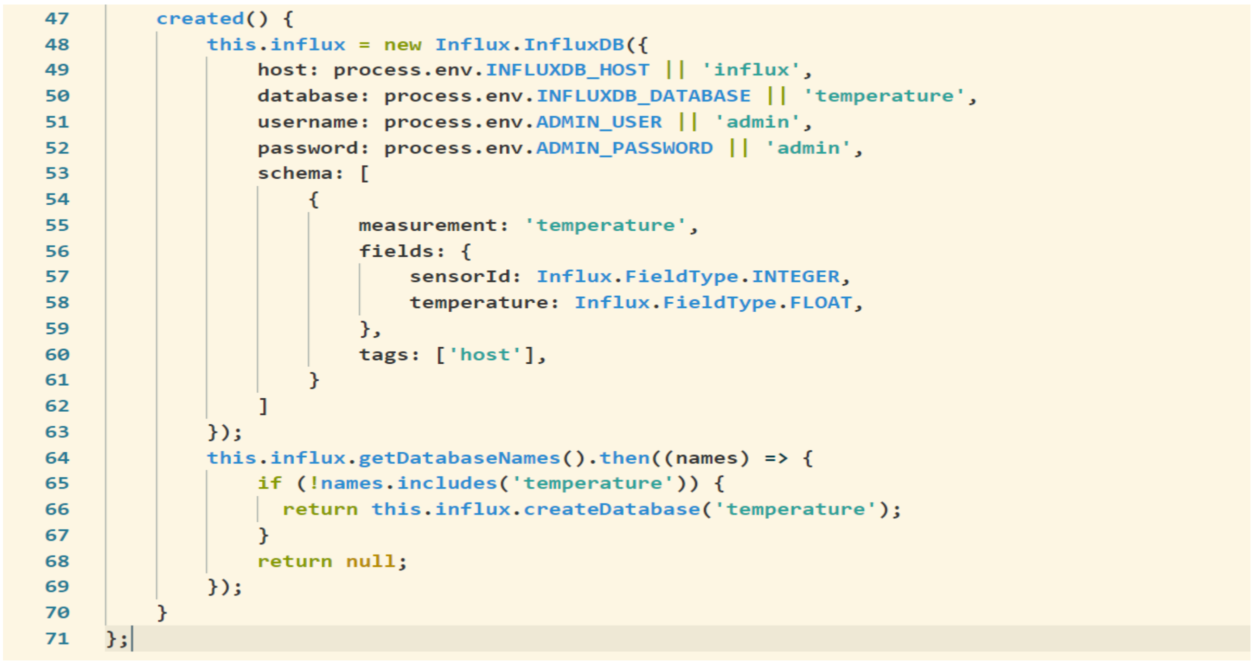

serializer.service.js ×
services > serializer.service.js > <unknown> > events > "temperature.read" > handler

```
"use strict";

const Influx = require('influx');

module.exports = {
    name: "serializer",
    actions: {
        read: {
            params: {
                sensorId: { type: "number" }
            },
            async handler(ctx) {
                try {
                    const res = await this.influx.query(
                        `select * from temperature where sensorId=${ctx.params.sensorId}`
                    );
                    return res;
                }
                catch(err) {
                    console.log(err);
                    return null;
                }
            }
        },
    },
```



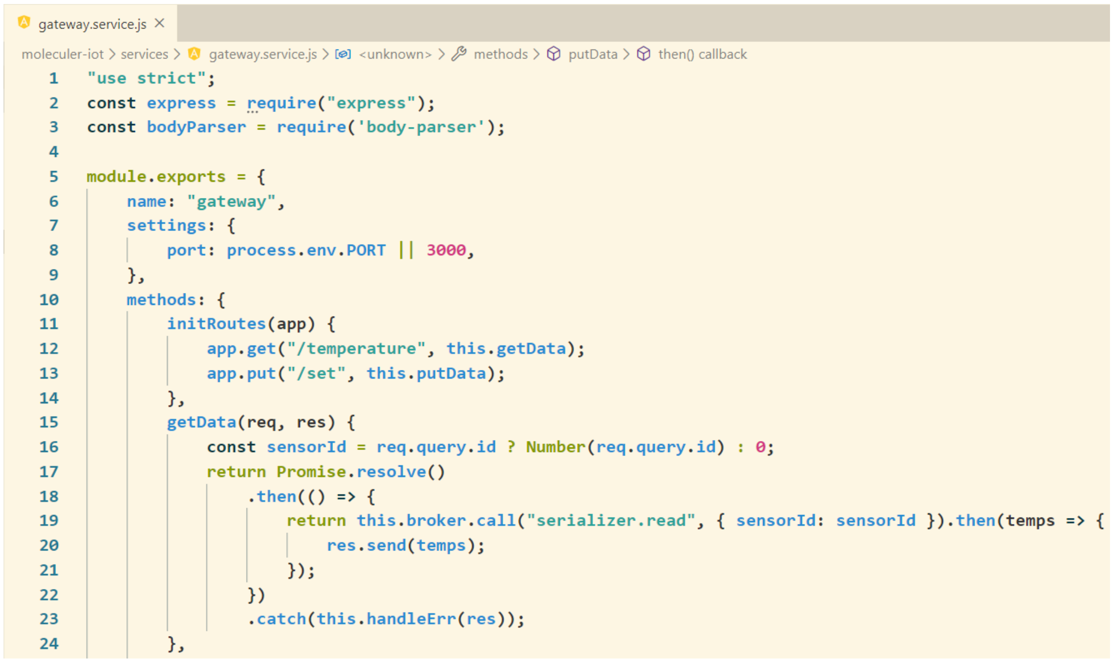
```
    events: {
        "temperature.read": {
            group: "other",
            handler(payload) {
                console.log(
                    'Recieved "temperature.read" event in serializer service with payload: ',
                    payload
                );
                this.influx.writePoints([
                    {
                        measurement: 'temperature',
                        fields: {
                            temperature: payload.temperature,
                            sensorId: payload.sensorId
                        },
                        time: payload.timestamp
                    }
                ]);
            }
        }
    },
```

Povezivanje sa bazom temperature – korišćenje influx modula za node.js (Moleculer nema kreiran adapter za InfluxDB). Kao host je naveden influx (naziv koji će se isto naći i u docker-compose.yml fajlu koji će biti prikazan kasnije). Polja koja će se upisivati pored vremena, su id senzora i vrednost temperature. Potrebno je pozvati i funkciju za čitanje imena baza podataka i ukoliko ne postoji temperature baza, potrebno je kreirati je.

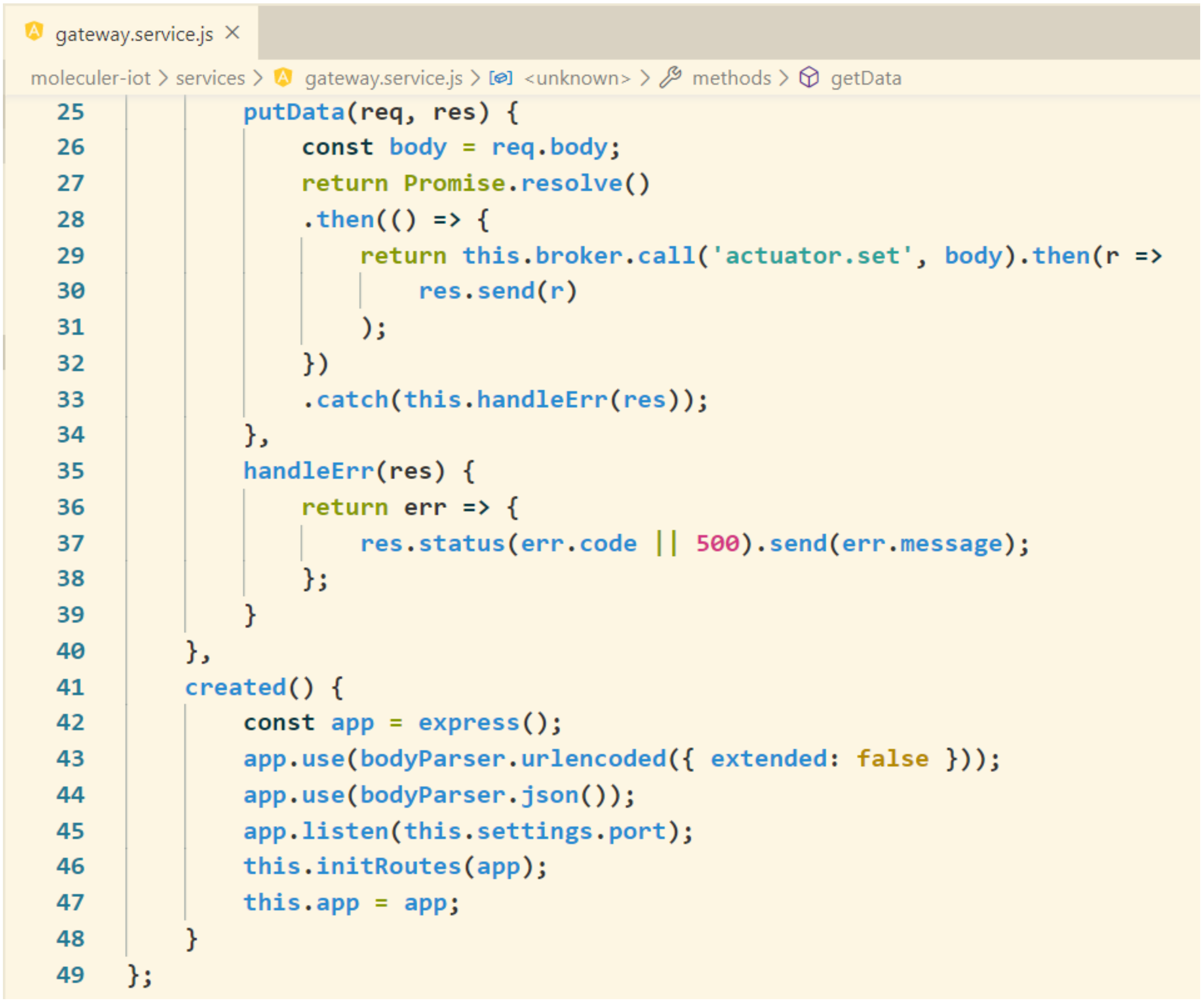


```
    created() {
        this.influx = new Influx.InfluxDB({
            host: process.env.INFLUXDB_HOST || 'influx',
            database: process.env.INFLUXDB_DATABASE || 'temperature',
            username: process.env.ADMIN_USER || 'admin',
            password: process.env.ADMIN_PASSWORD || 'admin',
            schema: [
                {
                    measurement: 'temperature',
                    fields: {
                        sensorId: Influx.FieldType.INTEGER,
                        temperature: Influx.FieldType.FLOAT,
                    },
                    tags: ['host'],
                }
            ]
        });
        this.influx.getDatabaseNames().then((names) => {
            if (!names.includes('temperature')) {
              return this.influx.createDatabase('temperature');
            }
            return null;
        });
    }
};
```

gateway.service.js – rute GET /temperature i PUT /set


gateway.service.js ×
moleculer-iot > services > gateway.service.js > <unknown> > methods > putData > then() callback

```
"use strict";
const express = require("express");
const bodyParser = require('body-parser');

module.exports = {
    name: "gateway",
    settings: {
        port: process.env.PORT || 3000,
    },
    methods: {
        initRoutes(app) {
            app.get("/temperature", this.getData);
            app.put("/set", this.putData);
        },
        getData(req, res) {
            const sensorId = req.query.id ? Number(req.query.id) : 0;
            return Promise.resolve()
                .then(() => {
                    return this.broker.call("serializer.read", { sensorId: sensorId }).then(temps => {
                        res.send(temps);
                    });
                })
                .catch(this.handleErr(res));
        },
```

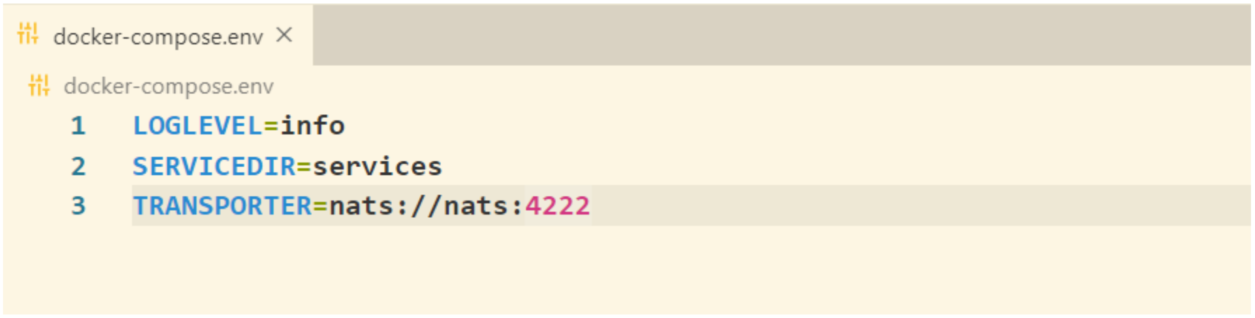


gateway.service.js ×
moleculer-iot > services > gateway.service.js > <unknown> > methods > getData

```
        putData(req, res) {
            const body = req.body;
            return Promise.resolve()
            .then(() => {
                return this.broker.call('actuator.set', body).then(r =>
                    res.send(r)
                );
            })
            .catch(this.handleErr(res));
        },
        handleErr(res) {
            return err => {
                res.status(err.code || 500).send(err.message);
            };
        }
    },
    created() {
        const app = express();
        app.use(bodyParser.urlencoded({ extended: false }));
        app.use(bodyParser.json());
        app.listen(this.settings.port);
        this.initRoutes(app);
        this.app = app;
    }
};
```

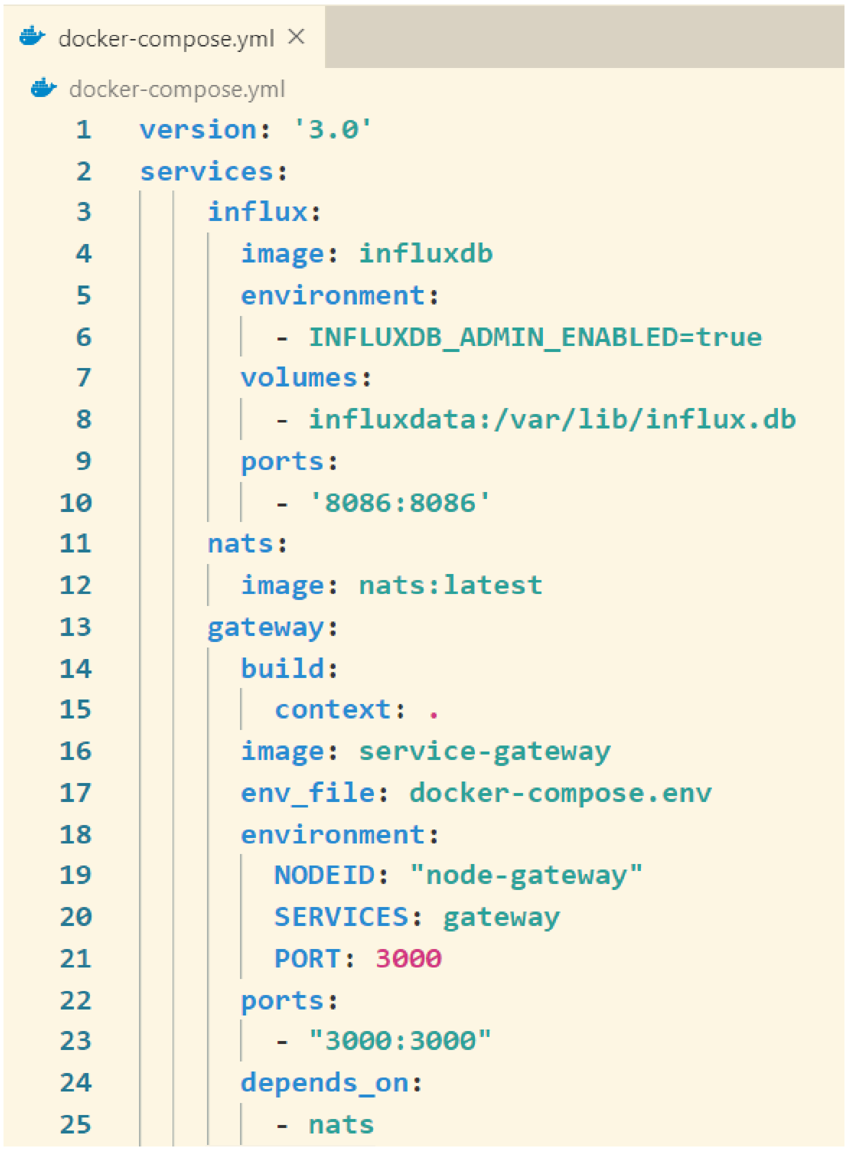
Docker

docker-compose.env – treba odrediti da je transporter **nats**

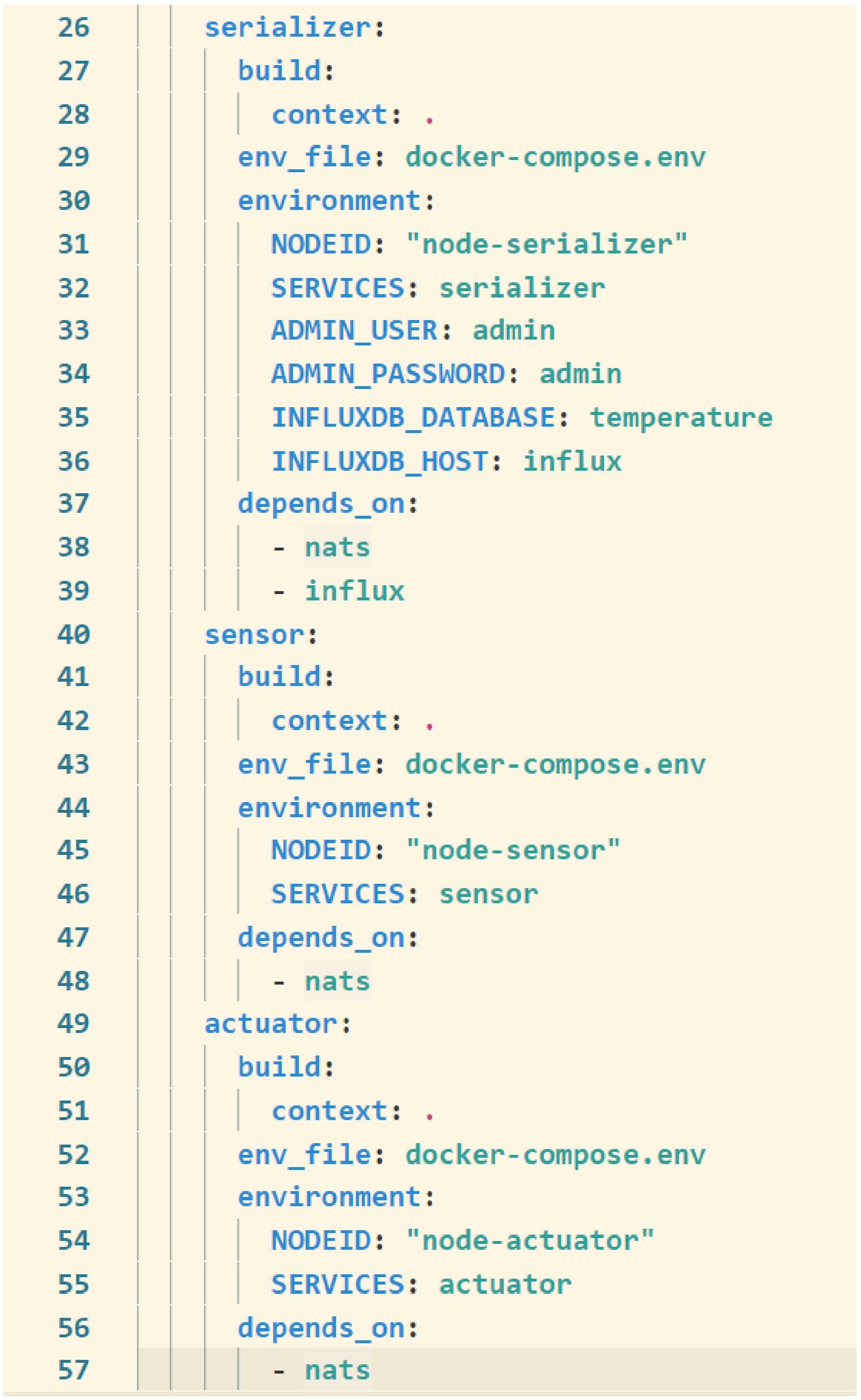


```
LOGLEVEL=info
SERVICEDIR=services
TRANSPORTER=nats://nats:4222
```

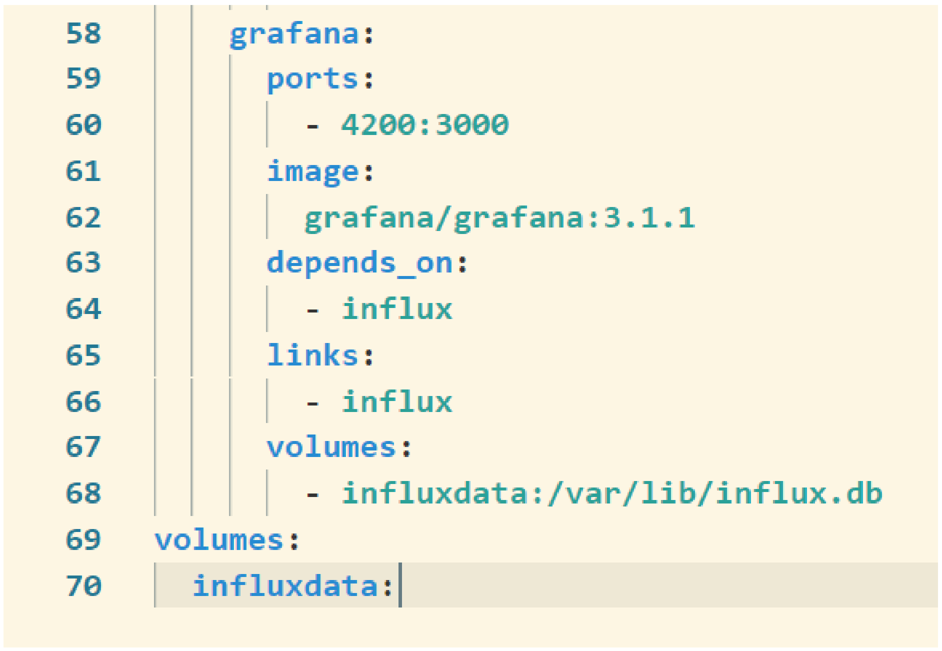
docker-compose.yml



```yaml
version: '3.0'
services:
    influx:
      image: influxdb
      environment:
        - INFLUXDB_ADMIN_ENABLED=true
      volumes:
        - influxdata:/var/lib/influx.db
      ports:
        - '8086:8086'
    nats:
      image: nats:latest
    gateway:
      build:
        context: .
      image: service-gateway
      env_file: docker-compose.env
      environment:
        NODEID: "node-gateway"
        SERVICES: gateway
        PORT: 3000
      ports:
        - "3000:3000"
      depends_on:
        - nats
```



```yaml
    serializer:
      build:
        context: .
      env_file: docker-compose.env
      environment:
        NODEID: "node-serializer"
        SERVICES: serializer
        ADMIN_USER: admin
        ADMIN_PASSWORD: admin
        INFLUXDB_DATABASE: temperature
        INFLUXDB_HOST: influx
      depends_on:
        - nats
        - influx
    sensor:
      build:
        context: .
      env_file: docker-compose.env
      environment:
        NODEID: "node-sensor"
        SERVICES: sensor
      depends_on:
        - nats
    actuator:
      build:
        context: .
      env_file: docker-compose.env
      environment:
        NODEID: "node-actuator"
        SERVICES: actuator
      depends_on:
        - nats
```



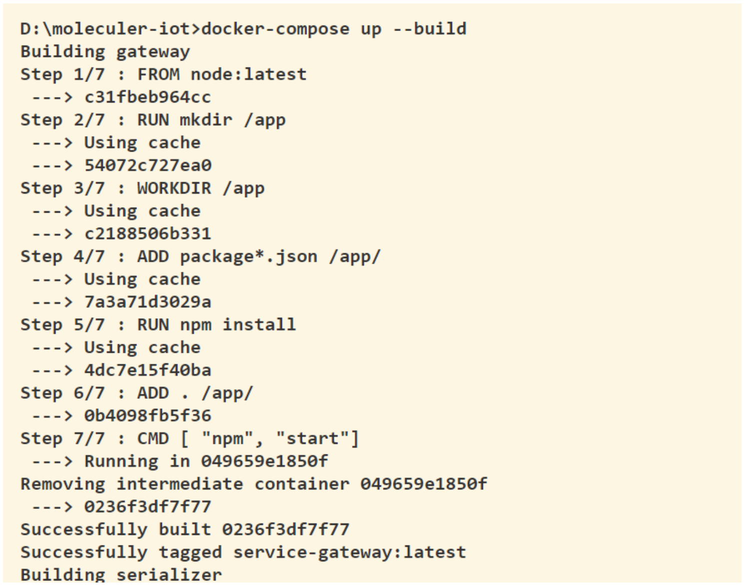
```
    grafana:
      ports:
        - 4200:3000
      image:
        grafana/grafana:3.1.1
      depends_on:
        - influx
      links:
        - influx
      volumes:
        - influxdata:/var/lib/influx.db
volumes:
  influxdata:
```

Pored kontejnera za naše mikroservise, potrebni su sledeći kontejneri:

1. nats
2. influx – baza na osnovu slike influxdb
3. grafana – na portu 4200 će se otvoriti interfejs ka Grafani, gde će biti vizuelizovani podaci iz influx baze, zato kažemo da zavisi od influx.

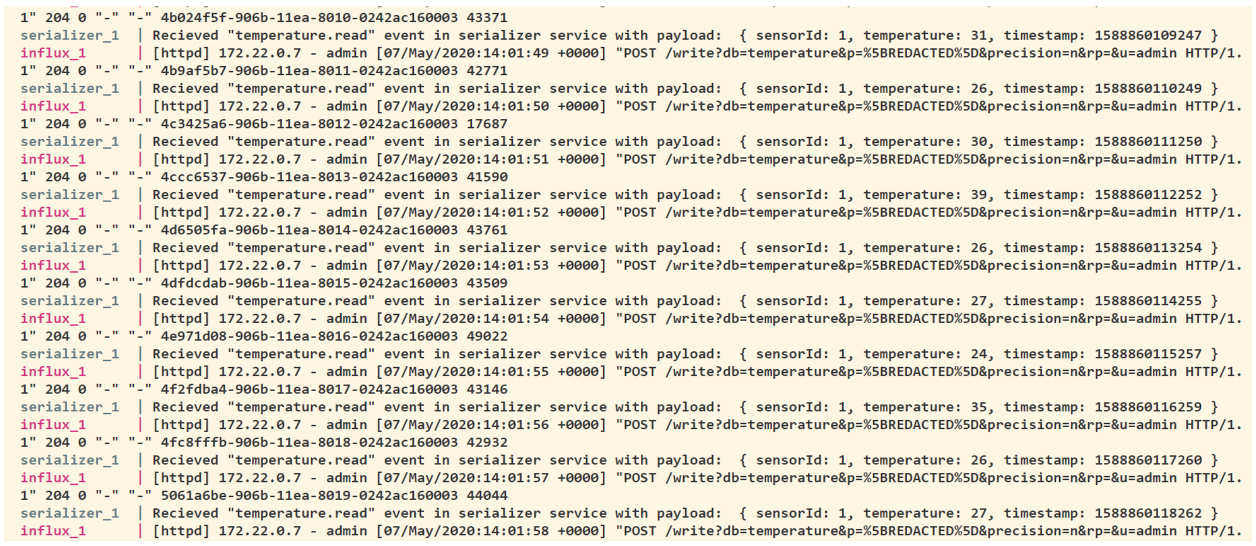
Pokrećemo naše kontejnere korišćenjem naredbe ***docker-compose up –build.***

Početak izvršenja naredbe:



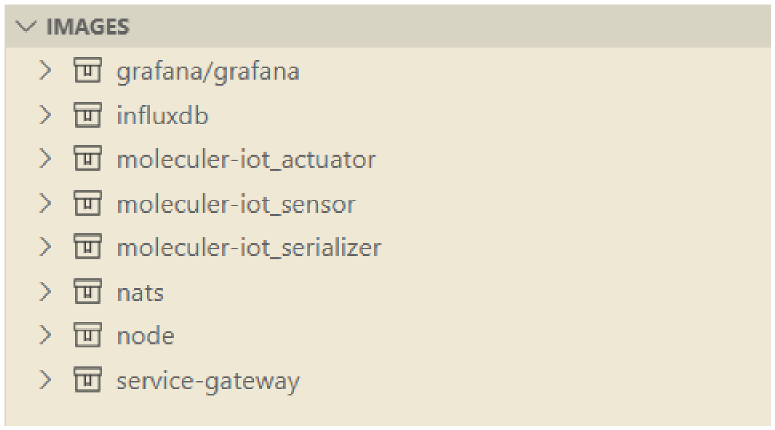
```
D:\moleculer-iot>docker-compose up --build
Building gateway
Step 1/7 : FROM node:latest
 ---> c31fbeb964cc
Step 2/7 : RUN mkdir /app
 ---> Using cache
 ---> 54072c727ea0
Step 3/7 : WORKDIR /app
 ---> Using cache
 ---> c2188506b331
Step 4/7 : ADD package*.json /app/
 ---> Using cache
 ---> 7a3a71d3029a
Step 5/7 : RUN npm install
 ---> Using cache
 ---> 4dc7e15f40ba
Step 6/7 : ADD . /app/
 ---> 0b4098fb5f36
Step 7/7 : CMD [ "npm", "start"]
 ---> Running in 049659e1850f
Removing intermediate container 049659e1850f
 ---> 0236f3df7f77
Successfully built 0236f3df7f77
Successfully tagged service-gateway:latest
Building serializer
```

Upisivanje merenja u bazu



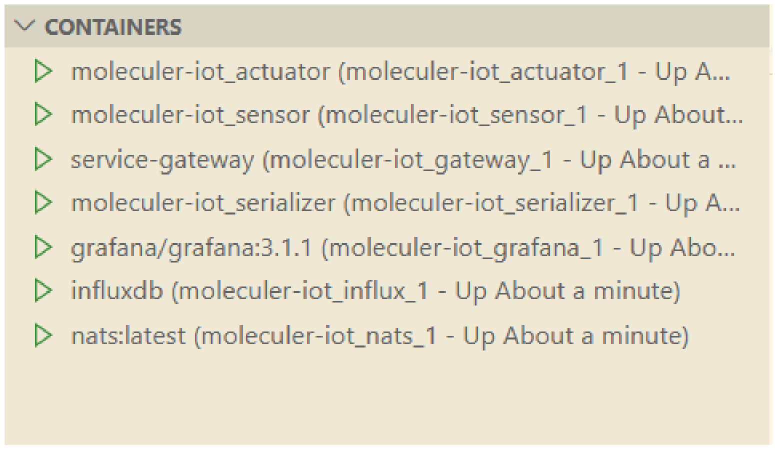
```
1" 204 0 "-" "-" 4b024f5f-906b-11ea-8010-0242ac160003 43371
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 31, timestamp: 1588860109247 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:49 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4b9af5b7-906b-11ea-8011-0242ac160003 42771
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 26, timestamp: 1588860110249 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:50 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4c3425a6-906b-11ea-8012-0242ac160003 17687
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 30, timestamp: 1588860111250 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:51 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4ccc6537-906b-11ea-8013-0242ac160003 41590
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 39, timestamp: 1588860112252 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:52 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4d6505fa-906b-11ea-8014-0242ac160003 43761
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 26, timestamp: 1588860113254 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:53 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4dfdcdab-906b-11ea-8015-0242ac160003 43509
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 27, timestamp: 1588860114255 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:54 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4e971d08-906b-11ea-8016-0242ac160003 49022
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 24, timestamp: 1588860115257 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:55 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4f2fdba4-906b-11ea-8017-0242ac160003 43146
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 35, timestamp: 1588860116259 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:56 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 4fc8fffb-906b-11ea-8018-0242ac160003 42932
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 26, timestamp: 1588860117260 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:57 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
1" 204 0 "-" "-" 5061a6be-906b-11ea-8019-0242ac160003 44044
serializer_1  | Recieved "temperature.read" event in serializer service with payload:  { sensorId: 1, temperature: 27, timestamp: 1588860118262 }
influx_1      | [httpd] 172.22.0.7 - admin [07/May/2020:14:01:58 +0000] "POST /write?db=temperature&p=%5BREDACTED%5D&precision=n&rp=&u=admin HTTP/1.
```

docker slike



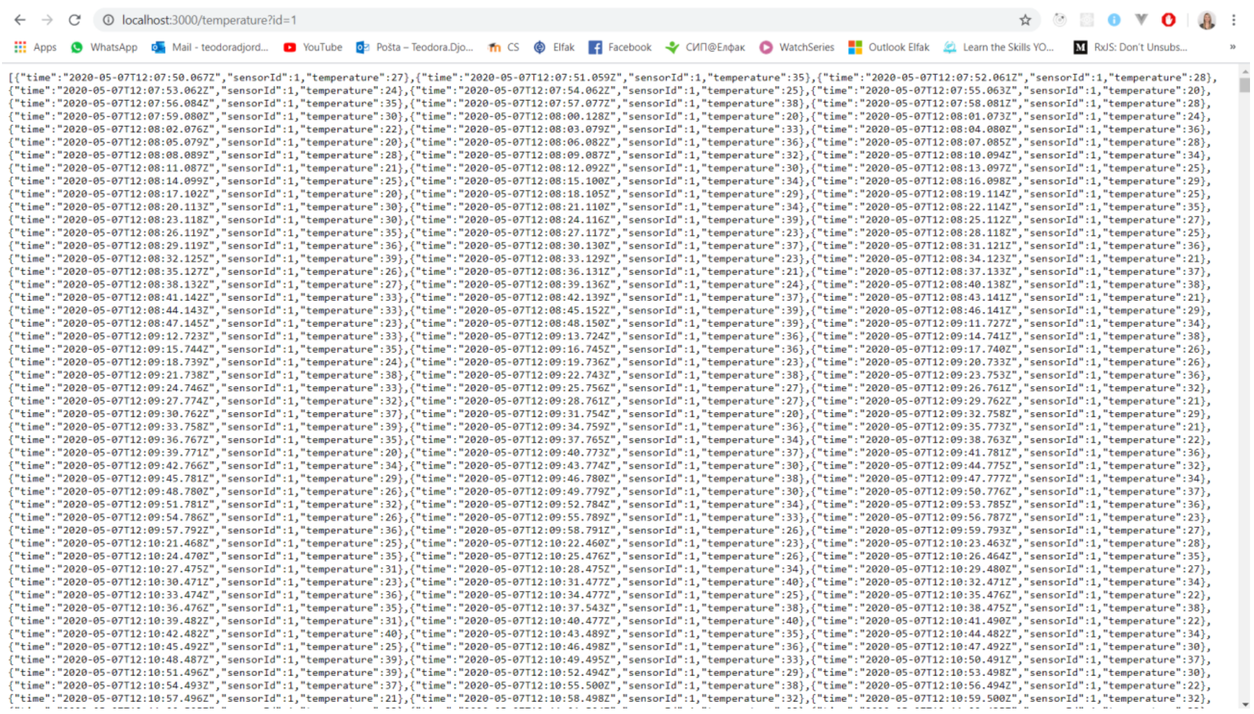
```
IMAGES
  grafana/grafana
  influxdb
  moleculer-iot_actuator
  moleculer-iot_sensor
  moleculer-iot_serializer
  nats
  node
  service-gateway
```

docker kontejneri



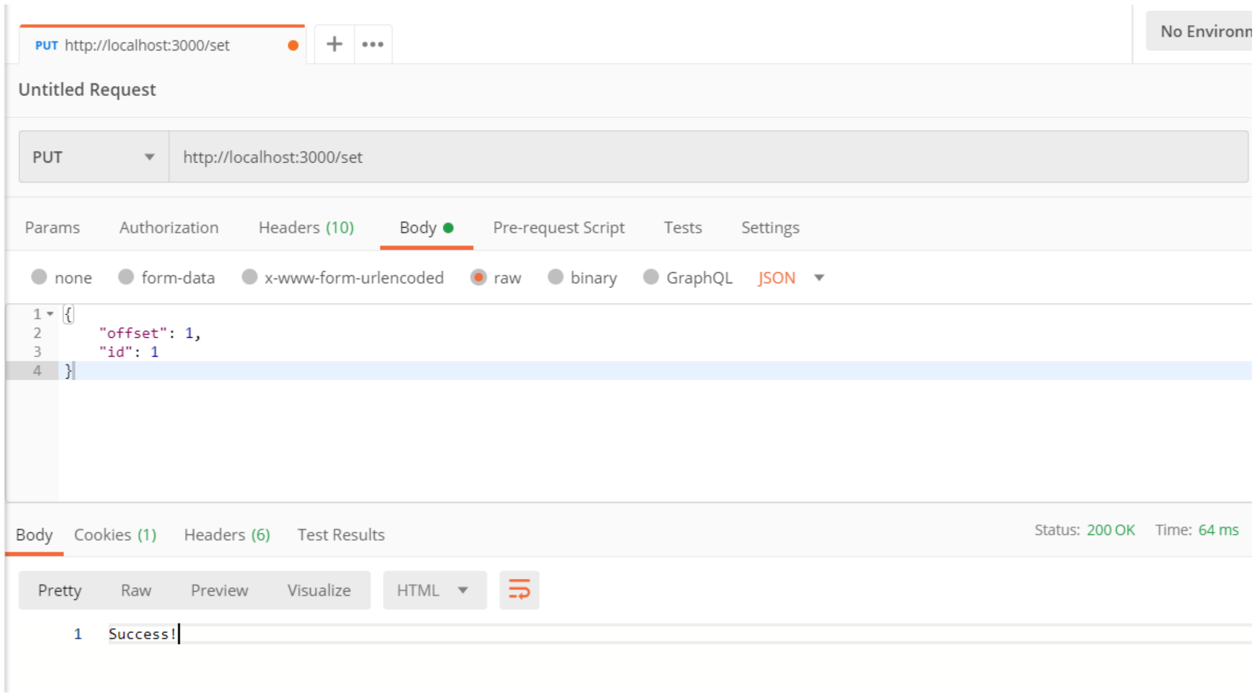
```
CONTAINERS
  moleculer-iot_actuator (moleculer-iot_actuator_1 - Up A...
  moleculer-iot_sensor (moleculer-iot_sensor_1 - Up About...
  service-gateway (moleculer-iot_gateway_1 - Up About a ...
  moleculer-iot_serializer (moleculer-iot_serializer_1 - Up A...
  grafana/grafana:3.1.1 (moleculer-iot_grafana_1 - Up Abo...
  influxdb (moleculer-iot_influx_1 - Up About a minute)
  nats:latest (moleculer-iot_nats_1 - Up About a minute)
```

GET na /temperature za senzor 1



```
[{"time":"2020-05-07T12:07:50.067Z","sensorId":1,"temperature":27},{"time":"2020-05-07T12:07:51.059Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:07:52.061Z","sensorId":1,"temperature":28},
{"time":"2020-05-07T12:07:53.062Z","sensorId":1,"temperature":24},{"time":"2020-05-07T12:07:54.062Z","sensorId":1,"temperature":25},{"time":"2020-05-07T12:07:55.063Z","sensorId":1,"temperature":20},
{"time":"2020-05-07T12:07:56.084Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:07:57.077Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:07:58.081Z","sensorId":1,"temperature":28},
{"time":"2020-05-07T12:07:59.080Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:08:00.128Z","sensorId":1,"temperature":20},{"time":"2020-05-07T12:08:01.073Z","sensorId":1,"temperature":24},
{"time":"2020-05-07T12:08:02.076Z","sensorId":1,"temperature":22},{"time":"2020-05-07T12:08:03.079Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:08:04.080Z","sensorId":1,"temperature":36},
{"time":"2020-05-07T12:08:05.079Z","sensorId":1,"temperature":20},{"time":"2020-05-07T12:08:06.082Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:08:07.085Z","sensorId":1,"temperature":28},
{"time":"2020-05-07T12:08:08.089Z","sensorId":1,"temperature":28},{"time":"2020-05-07T12:08:09.087Z","sensorId":1,"temperature":32},{"time":"2020-05-07T12:08:10.094Z","sensorId":1,"temperature":34},
{"time":"2020-05-07T12:08:11.087Z","sensorId":1,"temperature":21},{"time":"2020-05-07T12:08:12.092Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:08:13.097Z","sensorId":1,"temperature":25},
{"time":"2020-05-07T12:08:14.099Z","sensorId":1,"temperature":25},{"time":"2020-05-07T12:08:15.100Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:08:16.098Z","sensorId":1,"temperature":29},
{"time":"2020-05-07T12:08:17.102Z","sensorId":1,"temperature":20},{"time":"2020-05-07T12:08:18.105Z","sensorId":1,"temperature":29},{"time":"2020-05-07T12:08:19.114Z","sensorId":1,"temperature":25},
{"time":"2020-05-07T12:08:20.113Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:08:21.110Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:08:22.114Z","sensorId":1,"temperature":35},
{"time":"2020-05-07T12:08:23.118Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:08:24.116Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:08:25.112Z","sensorId":1,"temperature":27},
{"time":"2020-05-07T12:08:26.119Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:08:27.117Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:08:28.118Z","sensorId":1,"temperature":25},
{"time":"2020-05-07T12:08:29.119Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:08:30.130Z","sensorId":1,"temperature":37},{"time":"2020-05-07T12:08:31.121Z","sensorId":1,"temperature":36},
{"time":"2020-05-07T12:08:32.125Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:08:33.129Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:08:34.123Z","sensorId":1,"temperature":21},
{"time":"2020-05-07T12:08:35.127Z","sensorId":1,"temperature":26},{"time":"2020-05-07T12:08:36.131Z","sensorId":1,"temperature":21},{"time":"2020-05-07T12:08:37.133Z","sensorId":1,"temperature":37},
{"time":"2020-05-07T12:08:38.132Z","sensorId":1,"temperature":27},{"time":"2020-05-07T12:08:39.136Z","sensorId":1,"temperature":24},{"time":"2020-05-07T12:08:40.138Z","sensorId":1,"temperature":38},
{"time":"2020-05-07T12:08:41.142Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:08:42.139Z","sensorId":1,"temperature":37},{"time":"2020-05-07T12:08:43.141Z","sensorId":1,"temperature":21},
{"time":"2020-05-07T12:08:44.143Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:08:45.152Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:08:46.141Z","sensorId":1,"temperature":29},
{"time":"2020-05-07T12:08:47.145Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:08:48.150Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:09:11.727Z","sensorId":1,"temperature":34},
{"time":"2020-05-07T12:09:12.723Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:09:13.724Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:09:14.741Z","sensorId":1,"temperature":38},
{"time":"2020-05-07T12:09:15.744Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:09:16.745Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:09:17.740Z","sensorId":1,"temperature":26},
{"time":"2020-05-07T12:09:18.739Z","sensorId":1,"temperature":24},{"time":"2020-05-07T12:09:19.736Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:09:20.733Z","sensorId":1,"temperature":26},
{"time":"2020-05-07T12:09:21.738Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:09:22.743Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:09:23.753Z","sensorId":1,"temperature":36},
{"time":"2020-05-07T12:09:24.746Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:09:25.756Z","sensorId":1,"temperature":27},{"time":"2020-05-07T12:09:26.761Z","sensorId":1,"temperature":32},
{"time":"2020-05-07T12:09:27.774Z","sensorId":1,"temperature":32},{"time":"2020-05-07T12:09:28.761Z","sensorId":1,"temperature":27},{"time":"2020-05-07T12:09:29.762Z","sensorId":1,"temperature":21},
{"time":"2020-05-07T12:09:30.762Z","sensorId":1,"temperature":37},{"time":"2020-05-07T12:09:31.754Z","sensorId":1,"temperature":20},{"time":"2020-05-07T12:09:32.758Z","sensorId":1,"temperature":29},
{"time":"2020-05-07T12:09:33.758Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:09:34.759Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:09:35.773Z","sensorId":1,"temperature":21},
{"time":"2020-05-07T12:09:36.767Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:09:37.765Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:09:38.763Z","sensorId":1,"temperature":22},
{"time":"2020-05-07T12:09:39.771Z","sensorId":1,"temperature":20},{"time":"2020-05-07T12:09:40.773Z","sensorId":1,"temperature":37},{"time":"2020-05-07T12:09:41.781Z","sensorId":1,"temperature":36},
{"time":"2020-05-07T12:09:42.766Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:09:43.774Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:09:44.775Z","sensorId":1,"temperature":32},
{"time":"2020-05-07T12:09:45.781Z","sensorId":1,"temperature":29},{"time":"2020-05-07T12:09:46.780Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:09:47.777Z","sensorId":1,"temperature":34},
{"time":"2020-05-07T12:09:48.780Z","sensorId":1,"temperature":26},{"time":"2020-05-07T12:09:49.779Z","sensorId":1,"temperature":30},{"time":"2020-05-07T12:09:50.776Z","sensorId":1,"temperature":37},
{"time":"2020-05-07T12:09:51.781Z","sensorId":1,"temperature":32},{"time":"2020-05-07T12:09:52.784Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:09:53.785Z","sensorId":1,"temperature":36},
{"time":"2020-05-07T12:09:54.786Z","sensorId":1,"temperature":26},{"time":"2020-05-07T12:09:55.789Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:09:56.787Z","sensorId":1,"temperature":23},
{"time":"2020-05-07T12:09:57.792Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:09:58.791Z","sensorId":1,"temperature":26},{"time":"2020-05-07T12:09:59.793Z","sensorId":1,"temperature":27},
{"time":"2020-05-07T12:10:21.468Z","sensorId":1,"temperature":25},{"time":"2020-05-07T12:10:22.460Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:10:23.463Z","sensorId":1,"temperature":28},
{"time":"2020-05-07T12:10:24.470Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:10:25.476Z","sensorId":1,"temperature":26},{"time":"2020-05-07T12:10:26.464Z","sensorId":1,"temperature":35},
{"time":"2020-05-07T12:10:27.475Z","sensorId":1,"temperature":31},{"time":"2020-05-07T12:10:28.475Z","sensorId":1,"temperature":34},{"time":"2020-05-07T12:10:29.480Z","sensorId":1,"temperature":27},
{"time":"2020-05-07T12:10:30.471Z","sensorId":1,"temperature":23},{"time":"2020-05-07T12:10:31.477Z","sensorId":1,"temperature":40},{"time":"2020-05-07T12:10:32.471Z","sensorId":1,"temperature":34},
{"time":"2020-05-07T12:10:33.474Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:10:34.477Z","sensorId":1,"temperature":25},{"time":"2020-05-07T12:10:35.476Z","sensorId":1,"temperature":22},
{"time":"2020-05-07T12:10:36.476Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:10:37.543Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:10:38.475Z","sensorId":1,"temperature":38},
{"time":"2020-05-07T12:10:39.482Z","sensorId":1,"temperature":31},{"time":"2020-05-07T12:10:40.477Z","sensorId":1,"temperature":40},{"time":"2020-05-07T12:10:41.490Z","sensorId":1,"temperature":22},
{"time":"2020-05-07T12:10:42.482Z","sensorId":1,"temperature":40},{"time":"2020-05-07T12:10:43.489Z","sensorId":1,"temperature":35},{"time":"2020-05-07T12:10:44.482Z","sensorId":1,"temperature":34},
{"time":"2020-05-07T12:10:45.492Z","sensorId":1,"temperature":25},{"time":"2020-05-07T12:10:46.498Z","sensorId":1,"temperature":36},{"time":"2020-05-07T12:10:47.492Z","sensorId":1,"temperature":30},
{"time":"2020-05-07T12:10:48.487Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:10:49.495Z","sensorId":1,"temperature":33},{"time":"2020-05-07T12:10:50.491Z","sensorId":1,"temperature":37},
{"time":"2020-05-07T12:10:51.496Z","sensorId":1,"temperature":39},{"time":"2020-05-07T12:10:52.494Z","sensorId":1,"temperature":29},{"time":"2020-05-07T12:10:53.498Z","sensorId":1,"temperature":30},
{"time":"2020-05-07T12:10:54.493Z","sensorId":1,"temperature":37},{"time":"2020-05-07T12:10:55.500Z","sensorId":1,"temperature":38},{"time":"2020-05-07T12:10:56.494Z","sensorId":1,"temperature":22},
{"time":"2020-05-07T12:10:57.496Z","sensorId":1,"temperature":21},{"time":"2020-05-07T12:10:58.498Z","sensorId":1,"temperature":32},{"time":"2020-05-07T12:10:59.500Z","sensorId":1,"temperature":32},
```

PUT na /set



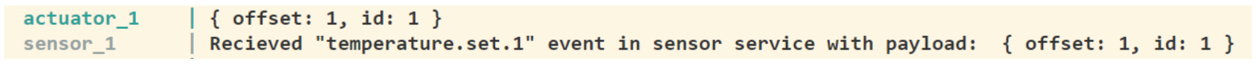
Untitled Request

PUT http://localhost:3000/set

```
{
    "offset": 1,
    "id": 1
}
```

Status: 200 OK Time: 64 ms

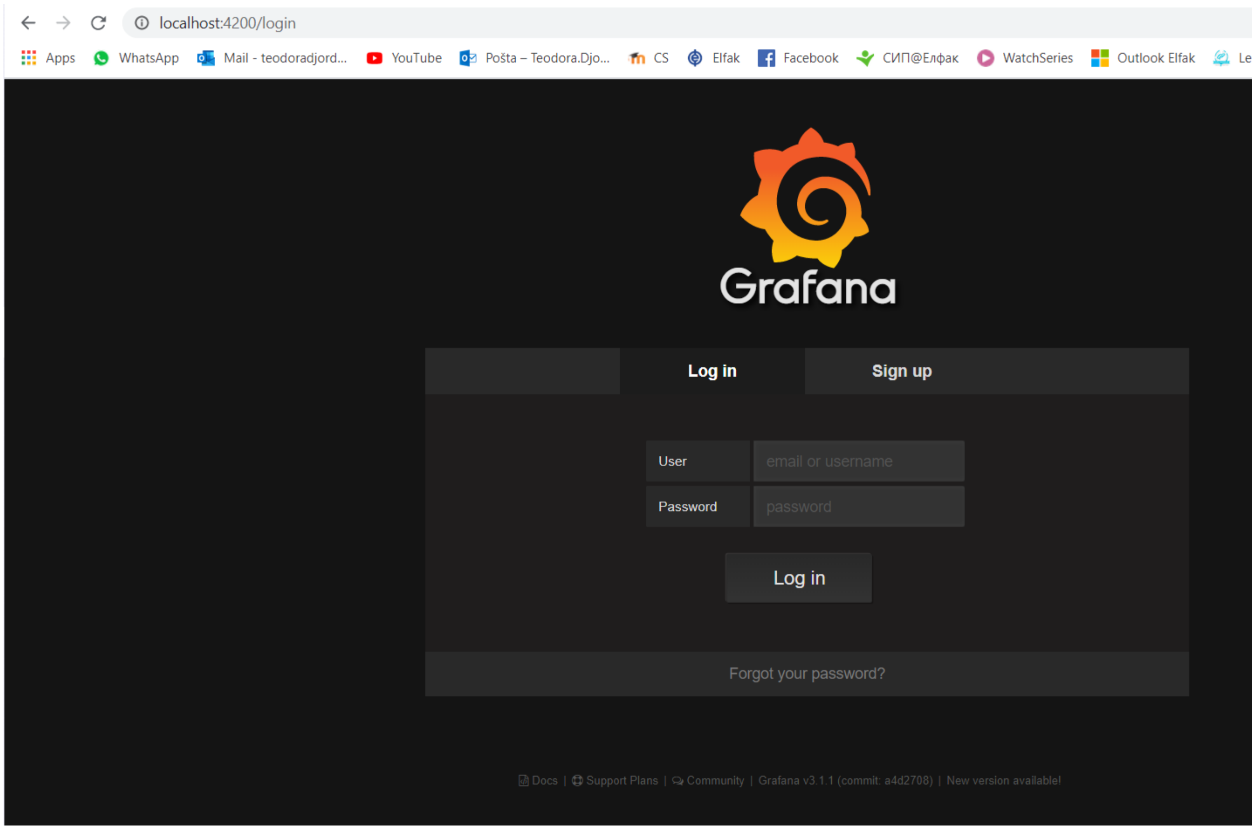
```
Success!
```



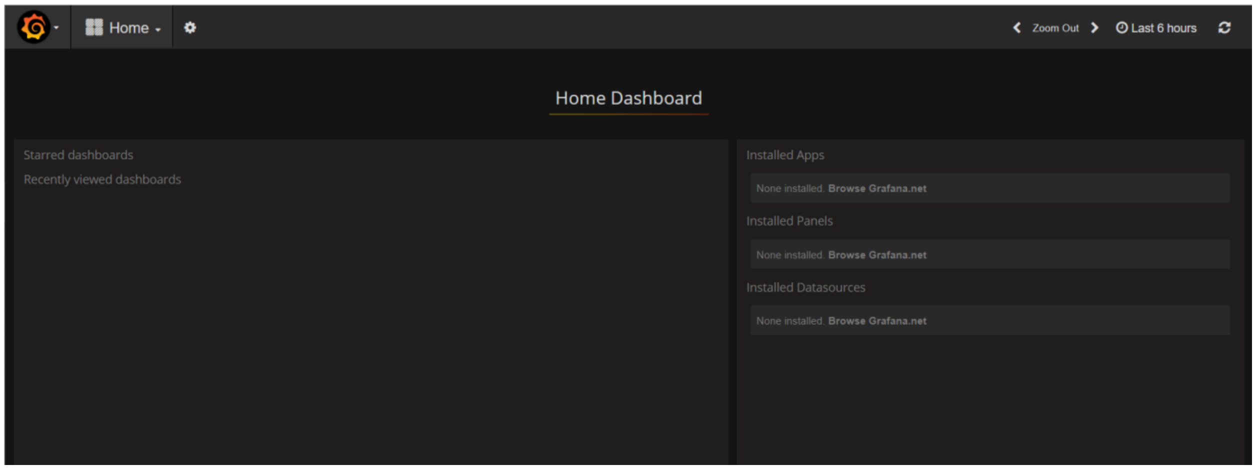
```
actuator_1    | { offset: 1, id: 1 }
sensor_1      | Recieved "temperature.set.1" event in sensor service with payload:  { offset: 1, id: 1 }
```

Grafana

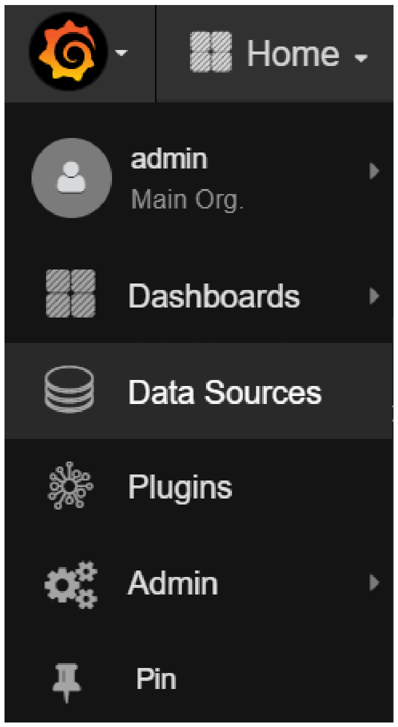
Na sledećoj slici je prikazana početna stranica na portu 4200 gde se treba ulogovati korišćenjem username-a admin i password-a admin.

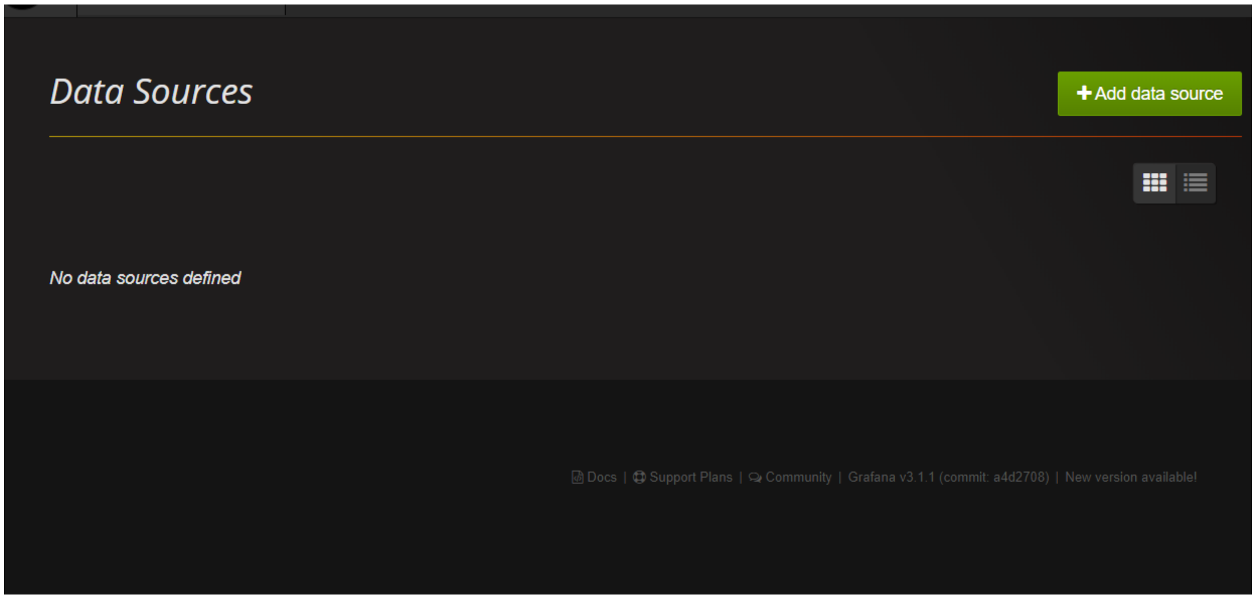


Prikaz stranice nakon logovanja


Home
Zoom Out
Last 6 hours
Home Dashboard
Starred dashboards
Recently viewed dashboards
Installed Apps
None installed. Browse Grafana.net
Installed Panels
None installed. Browse Grafana.net
Installed Datasources
None installed. Browse Grafana.net

Dodavanje data source-a je prikazano na narednim slikama. Prvo iz menija treba odabrati data sources stavku, a nakon toga kliknuti dugme add data source.

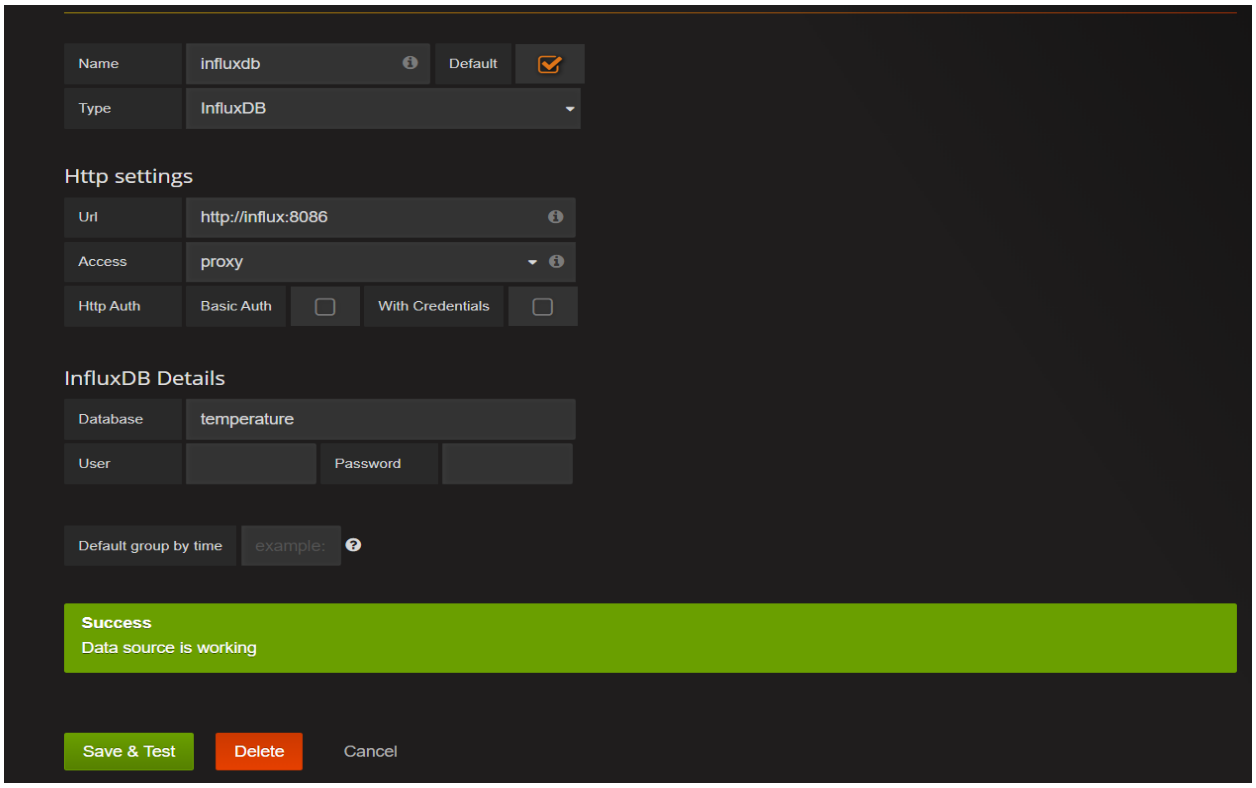




No data sources defined

Docs | Support Plans | Community | Grafana v3.1.1 (commit: a4d2708) | New version available!

Ime možemo da odaberemo proizvoljno, a kao type treba odabrati InfluxDB. Treba čekirati default dugme. U http settings delu treba odabrati http://influx:8086 – influx je ime koje smo zadali u docker-compose fajlu. Kao ime baze treba navesti temperature.



Name	influxdb	Default	✓
Type	InfluxDB		

Http settings

Url	http://influx:8086		
Access	proxy		
Http Auth	Basic Auth ☐	With Credentials ☐	

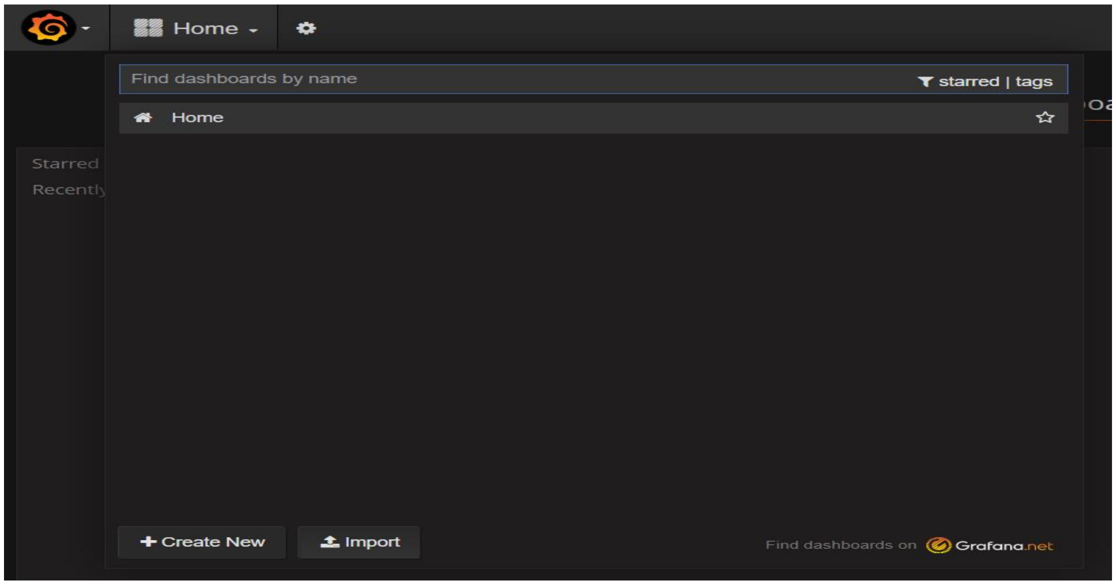
InfluxDB Details

Database	temperature		
User		Password	

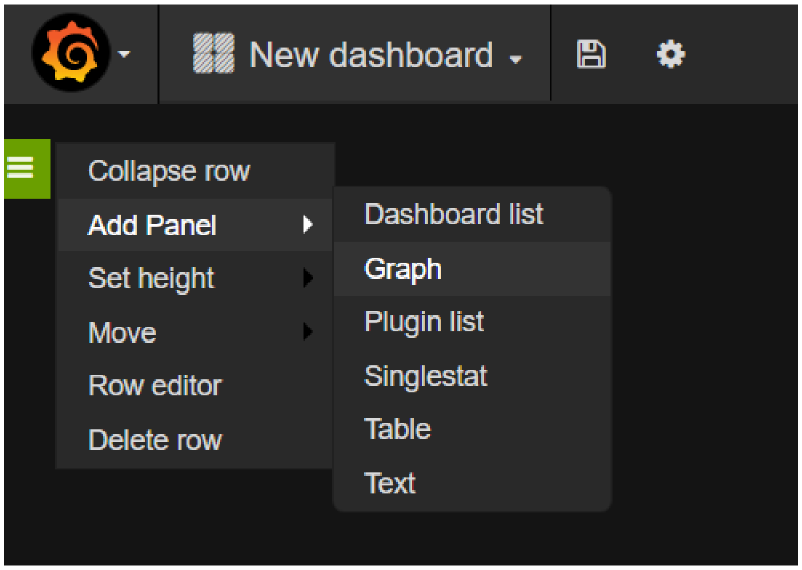
Default group by time example:

Success
Data source is working

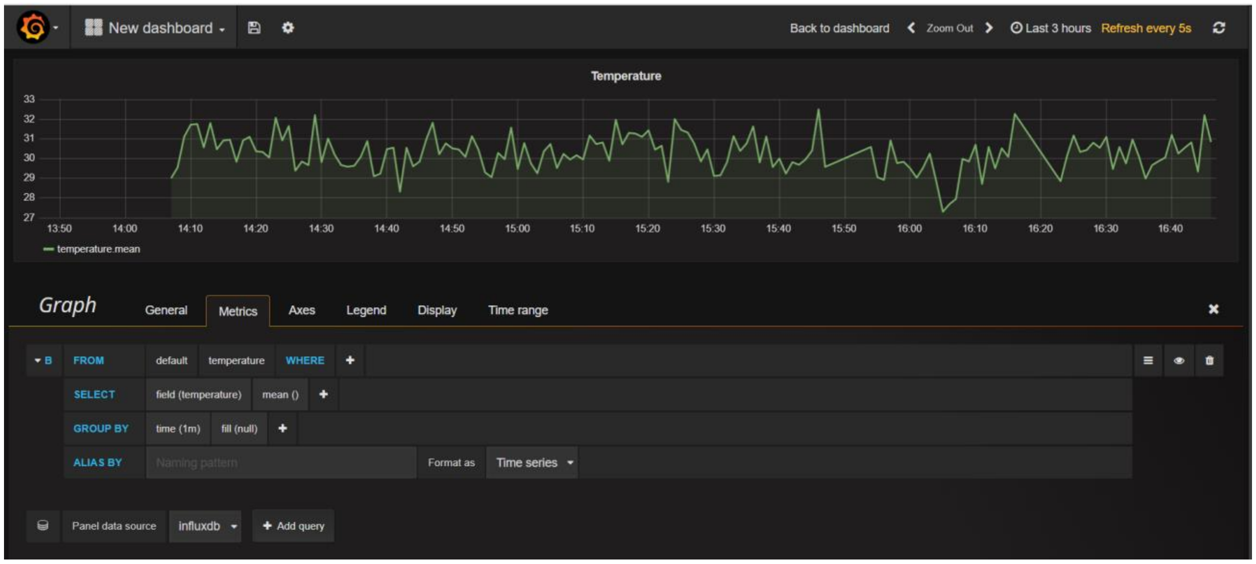
Sledeće u home delu treba kreirati novi dashboard klikom na dugme create new.



U ovom meniju treba odabrati graph.



Na sledećoj slici je prikazano kako treba podesiti prikaz podataka. Prvo u delu panel data source treba odabrati influxdb – data source koji smo kreirali, a nakon toga konstruisati upit koji će iz baze temperature da čita sve temperature grupisane po vremenu. Periode je moguće menjati. Takođe, u gornjem desnom uglu je moguće podesiti koliko često se osvežava i gledati uživo kako se temperatura menja dok je aplikacija aktivna.



B FROM default temperature WHERE +
SELECT field (temperature) mean () +
GROUP BY time (1m) fill (null) +
ALIAS BY Naming pattern Format as Time series
Panel data source influxdb + Add query

Github repozitorijum u kome se nalazi kod je moguće videti OVDE.