Jmeter and Java

The first point of entry in our system will be Jmeter, we plan to run the JMeter engine in java. At this stage, we use a pre-created test plan to then trigger it in java by using the StandardJMeterEngine package, and the method's init

```
public void test() throws IOException {


// JMeter Engine
StandardJMeterEngine jmeter = new StandardJMeterEngine();


// Initialize Properties, logging, locale, etc.
JMeterUtils.loadJMeterProperties("C:\\apache-jmeter-5.4.1\\bin\\jmeter.properties"); // /path/to/y
JMeterUtils.setJMeterHome("C:\\apache-jmeter-5.4.1"); // /path/to/your/jmeter
JMeterUtils.initLocale();

// Initialize JMeter SaveService
SaveService.loadProperties();

// Load existing .jmx Test Plan
//HashTree testPlanTree = SaveService.loadTree(new File("./src/test/resources/firstTestCraigGallen
HashTree testPlanTree = SaveService.loadTree(new File(".\\resource\\TestPlan.jmx")); // /path/to/y

Summariser summer = null;
String summariserName = JMeterUtils.getPropDefault("summariser.name", "summary");
if (summariserName.length() > 0) {
    summer = new Summariser(summariserName);
}


// Store execution results into a .jtl file
String logFile = "./target/standardTest" + Instant.now().toString().replace(":", "-") + ".jtl\"";
ResultCollector logger = new ResultCollector(summer);
logger.setFilename(logFile);
testPlanTree.add(testPlanTree.getArray()[0], logger);

// Run JMeter Test
jmeter.configure(testPlanTree);
jmeter.run();
```

The Test plan is linked to a Back listner, which is linked to an influxdb server running on the docker engine. All results that are produced by the Java class get pushed to the influxdb server

```
version: "3"
services:
  grafana:
    image: grafana/grafana
    container_name: grafana
    restart: always
    ports:
      - 3000:3000
    networks:
      - grafana_network
    volumes:
      - grafana_data:/var/lib/grafana
    depends_on:
      - influxdb


  influxdb:
    image: influxdb:1.8
    container_name: influxdb
    restart: always
    ports:
      - 8086:8086
    networks:
      - grafana_network
    volumes:
      - influxdb_data:/var/lib/influxdb
    environment:
      - INFLUXDB_DB=jmeter
      - INFLUXDB_USER=ICPuser
      - INFLUXDB_USER_PASSWORD=ICPpass
      - INFLUXDB_ADMIN_ENABLED=true
      - INFLUXDB_ADMIN_USER=admin
      - INFLUXDB_ADMIN_PASSWORD=admin
networks:
  grafana_network:
volumes:
  grafana_data:
  influxdb_data:
```

Docker Compose

This Docker compose is built to create Grafana as well as influxdb, using images found on Docker Hub. It then links the two together on the same network, to make connecting them together a lot easier. Volumes are given to both images to allow them to store data that's from JMeter.

Grafana

The Grafana image in this Docker Compose file can be found on the server IP it's running on, at Port 3000. The data is then stored at “var/lib/grafana”. It depends on the influxdb database to be running for it to initialise

Influxdb

The influxdb image uses its 1.8 version, so we can take advantage of looking at the data on the influx shell. It is linked to JMeter and Grafana at port 8086, on whatever server it's running on. The data is stored at var/lib/influxdb. The environment variables allow us to set the databases which the information will be stored in , in this case it;s JMeter. The databases can be accessed using the account details ICPuser and ICPpass

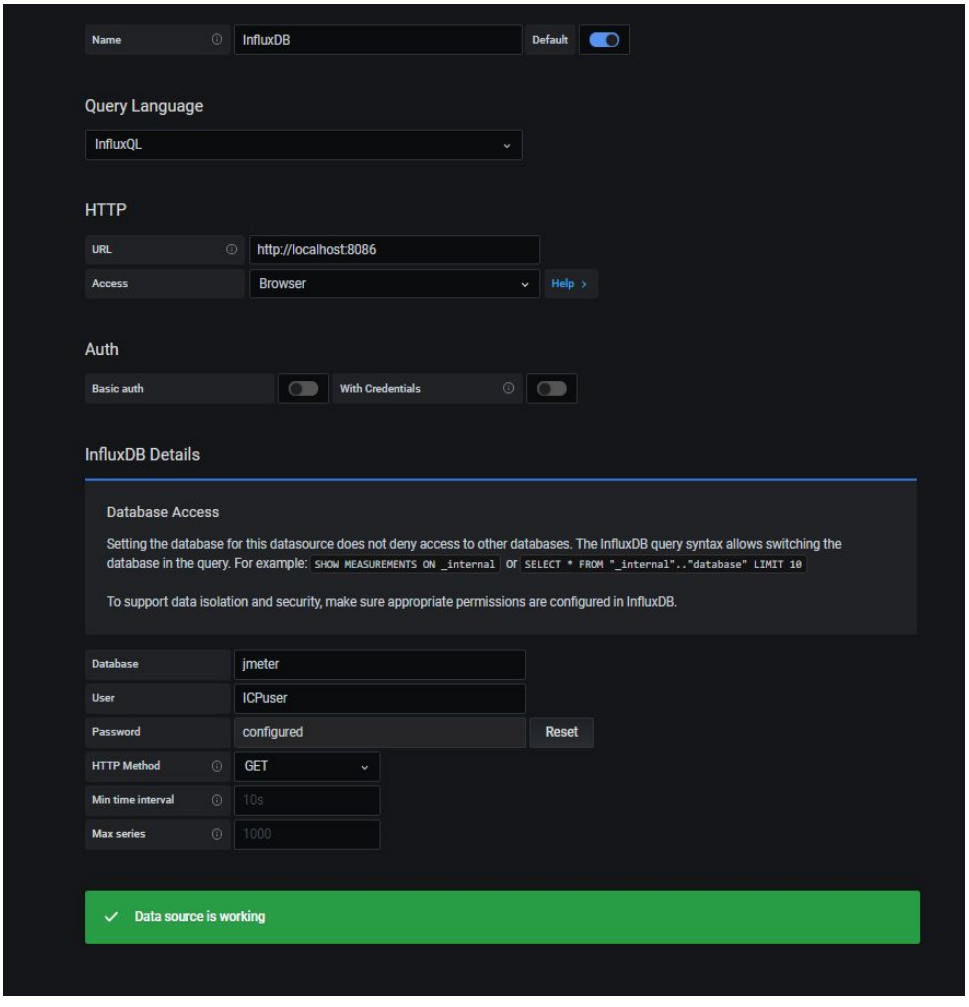
```
C:\Users\tebog\Documents\ICP\Docker>docker exec -it influxdb /bin/bash
root@7ec1a209eea0:/# which influx
/usr/bin/influx
root@7ec1a209eea0:/# cd usr/bin
root@7ec1a209eea0:/usr/bin# influx
Connected to http://localhost:8086 version 1.8.4
InfluxDB shell version: 1.8.4
> use jmeter
Using database jmeter
> select * from jmeter
```

We can access the container of influxdb with the command ***“docker exec -it influxdb /bin/bash”*** here we can make sure the data is being passed through by outputting all results found in the JMeter database (where we are sending all the information.

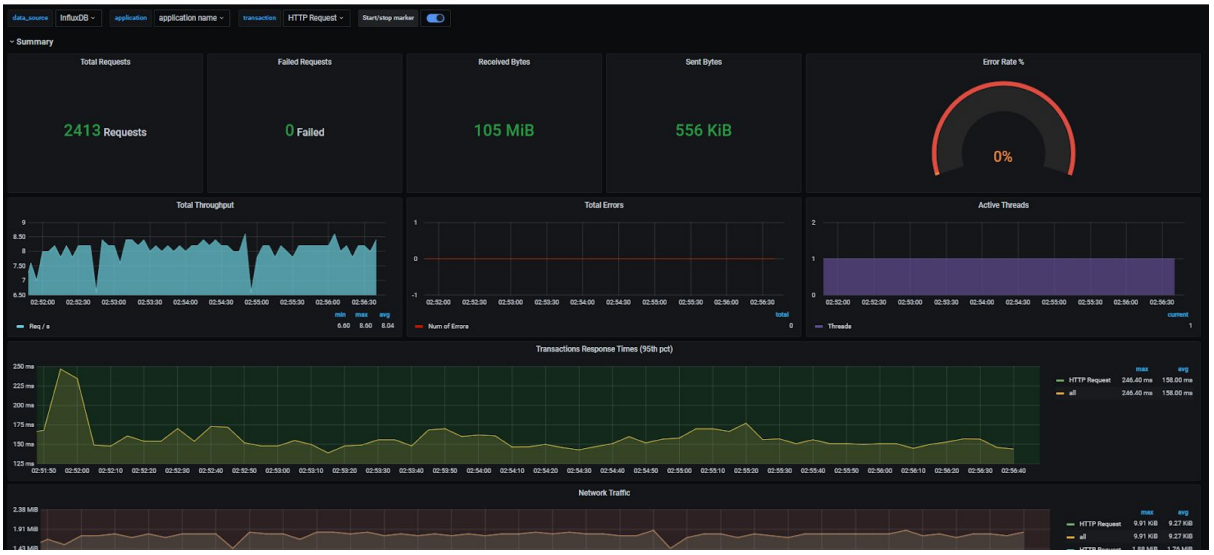
```
1614567128900000000 application name 123.24            40                    205             94      141.9           149             204.62999999999982
                    ok          HTTP Request
1614567128900000000 application name 123.24            40                    205             94      141.9           149             204.62999999999982 183
2360 9440           all         HTTP Request
1614567133899000000 application name 122.69            41                    168             100     138.70000000000002 147.74999999999994 167.92999999999995
                    ok          HTTP Request
1614567133899000000 application name 122.69            41                    168             100     138.70000000000002 147.74999999999994 167.92999999999995 187
8169 9676           all         HTTP Request
1614567133899000000 application name 122.69            41    0           123 168             100     138.70000000000002 147.74999999999994 167.92999999999995 187
8169 9676           all         all
1614567133899000000 application name                                  0           1     1         1
          1         internal
1614567138898000000 application name 124.81            39    0           117 211             93      144.8           160.64999999999992 210.7799999999999  178
6551 9204           all         all
1614567138898000000 application name                                  0           1     1         1
          1         internal
1614567138898000000 application name 124.81            39                    211             93      144.8           160.64999999999992 210.7799999999999
                    ok          HTTP Request
1614567138898000000 application name 124.81            39                    211             93      144.8           160.64999999999992 210.7799999999999  178
6551 9204           all         HTTP Request
1614567143895000000 application name 124.06            41                    211             88      144.9           153.89999999999998 210.7799999999999
                    ok          HTTP Request
1614567143895000000 application name 124.06            41                    211             88      144.9           153.89999999999998 210.7799999999999  187
8169 9676           all         HTTP Request
1614567143895000000 application name 124.06            41    0           123 211             88      144.9           153.89999999999998 210.7799999999999  187
8169 9676           all         all
1614567143895000000 application name                                  0           1     1         1
          1         internal
1614567148892000000 application name 124.49            39    0           117 383             88      144             153.84999999999997 381.049999999999   178
6551 9204           all         all
1614567148892000000 application name                                  0           1     1         1
          1         internal
1614567148892000000 application name 124.49            39                    383             88      144             153.84999999999997 381.049999999999
                    ok          HTTP Request
1614567148892000000 application name 124.49            39                    383             88      144             153.84999999999997 381.049999999999   178
6551 9204           all         HTTP Request
1614567153903000000 application name 124.01            41                    383             88      143.9           170.1499999999998  380.989999999999
                    ok          HTTP Request
1614567153903000000 application name 124.01            41                    383             88      143.9           170.1499999999998  380.989999999999   187
8169 9676           all         HTTP Request
1614567153903000000 application name 124.01            41    0           123 383             88      143.9           170.1499999999998  380.989999999999   187
8169 9676           all         all
1614567153903000000 application name                                  0           1     1         1
          1         internal
1614567158898000000 application name 123.13            41    0           123 383             89      141             153.84999999999997 381.029999999999   187
8169 9676           all         all
1614567158898000000 application name                                  0           1     1         1
          1         internal
1614567158898000000 application name 123.13            41                    383             89      141             153.84999999999997 381.029999999999   187
8169 9676           all         HTTP Request
1614567158898000000 application name 123.13            41                    383             89      141             153.84999999999997 381.029999999999
                    ok          HTTP Request
```

Grafana

This is how we linked influxdb to Grafana, using all the settings we had set for influxdb in the docker compose file.



Grafana Graph



This is the result of all the data from influxdb, being displayed in graph form in Grafana using metrics specially built for JMeter results.