# View MicroProfile metrics with Prometheus and Grafana

Rudy De Busscher

# Table of Contents

User manual for Atbash configuration.

# Introduction

This document is the steps in the demo of the presentation *Deploy, monitor, and take control of your Micro-Services with MicroProfile*.

It shows you how you can show the metrics in Grafana charts.

# Requirements

This demo is using docker images and containers, so a fully working docker environment is expected. The *docker* command needs to be executable, so you can try this command to see if everything is ok.

```
docker version
```

It will also download some Docker images from the internet, so a connection is also required.

# Getting the code / scripts

Download the code from the GitHub repository https://github.com/rdebusscher/mp-metrics-demo on your machine. This directory will be referred to as <**source_home**>.

To make it easier to work with the different docker containers, you can open 3 different terminal windows. they all need a specific base directory to work with them easily.

1. Terminal <**project**> with the directory <*source_home*> as the current directory`
2. Terminal <**prometheus**> with the directory <*source_home*>*/src/prometheus/docker* as the current directory
3. Terminal <**grafana**> which can have any directory as the current directory.

# The steps

## Create network

To make it easier to connect the different docker instances, create a specific network in docker with the following command.

```
docker network create demo-net
```

# Create Application image

For our tests, we are creating a specific image which is based on the official Payara Docker image where we add the WAR file with our application.

You can execute the following commands in the <**project**> terminal.

```
mvn clean package
```

And then to create the image

```
docker build -t wjax/service .
```

# Startup the image with Application.

Now that we have the Docker image, let start up a container with this image. ( <**project**> terminal but not required)

```
docker run -d -p 8080:8080 --name service --net demo-net wjax/service
```

The name *service* is here important as it is used in the Prometheus configuration. We have defined that it looks up the application through DNS by using the names *service*.

You can verify if everything is ok by calling the following URL in your browser

```
http://localhost:8080/monitoring/rest/hello
```

# Create Prometheus image

We are creating a special Docker image containing the Prometheus server (from Adam Bien's Dockland https://github.com/AdamBien/docklands/tree/master/prometheus)

Run the build image command from the **prometheus** terminal.

```
docker build -t wjax/prometheus .
```

# Startup the image with Prometheus

Now that we have Prometheus Docker image, let start the container with this image.

```
docker run -d -p 9090:9090 --name prometheus --net demo-net wjax/prometheus
```

You can verify if the connection with the application works with the following steps

1. Open the browser with the URL http://localhost:9090.
2. Select the metrics *vendor:system_cpu_load* in the drop-down.
3. Put some load on your machine with the badly written (on purpose) multi-threaded check with prime numbers on http://localhost:8080/monitoring/rest/prime.
4. You should see the spike in CPU usage after you have pressed the *execute* button.

# Start the Grafana instance

For the Grafana instance, no specific image or configuration is required. We can just run the default image and define the Prometheus instance within the GUI.

Within the **grafana** terminal, you can run the following command.

```
docker run -d --name grafana --net demo-net -p 3000:3000 grafana/grafana
```

# Lookup IP address of Prometheus

When we connect Grafana to Prometheus, it is best that we do this directly through the *demo-net* we created within docker. So we can find out this address of the Prometheus instance by running the following command

```
docker inspect demo-net
```

And look for the IP address of the Prometheus instance.

# Configure/use Grafana

When using Grafana, you can look up the usage on the internet as this is nothing specific for MicroProfile metrics. Here a few quick getting started things

- Open the browser with the URL *http://localhost:3000*
- Login with the default admin/admin username/password combination.
- Change the password to something personal.
- Add a data source to the Prometheus server, using that IP address that you discovered in the previous step
- Add a dashboard and graph, using the metric *vendor:system_cpu_load* as an example, the same as we used above.