**API Monitoring Setup**

**Required changes from Dev team:**

POM updates:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<!-- Micormeter core dependency -->

<dependency>

<groupId>io.micrometer</groupId>

<artifactId>micrometer-core</artifactId>

</dependency>

<!-- Micrometer Prometheus registry -->

<dependency>

<groupId>io.micrometer</groupId>

<artifactId>micrometer-registry-prometheus</artifactId>

</dependency>

##application.properties updates ::

management.endpoint.metrics.enabled=true

management.endpoint.prometheus.enabled=true

management.metrics.export.prometheus.enabled=true

management.endpoints.web.exposure.include=health,info,prometheus,metrics

**Prometheus installation and configuration:**

#Check and update the sestatus

* sestatus
* setenforce 0

#Install wget

* yum install wget -y

#Download the Prometheus package

* wget <https://github.com/prometheus/prometheus/releases/download/v2.18.1/prometheus-2.18.1.linux-amd64.tar.gz>

#Once the archive is downloaded, extract it using the below command

* tar -xvzf prometheus-2.18.1.linux-amd64.tar.gz

#We need a separate user and the group for the Prometheus

* groupadd --system prometheus
* useradd -s /sbin/nologin -r -g prometheus prometheus

#Setup Directories for Prometheus

#/var/lib/prometheus – Will be used to store Prometheus data

* mkdir -p /var/lib/prometheus

#/etc/prometheus – will be used to store configuration files

* mkdir -p /etc/prometheus

#Copy the Prometheus binary files to the directory where all the user libraries will be stored

* cp prometheus-2.18.1.linux-amd64/prometheus /usr/local/bin/
* cp prometheus-2.18.1.linux-amd64/promtool /usr/local/bin/

#Copying Prometheus Console and Console Libraries To The Prometheus Configuration Directory

* cp -r prometheus-2.18.1.linux-amd64/consoles /etc/prometheus/
* cp -r prometheus-2.18.1.linux-amd64/console\_libraries /etc/prometheus/
* cp prometheus-2.18.1.linux-amd64/prometheus.yml /etc/prometheus/

#Changing Directory Ownership

* chown -R prometheus:prometheus /etc/prometheus/ /var/lib/prometheus/
* chmod -R 775 /etc/prometheus/ /var/lib/prometheus/

#Setup Systemd File for Prometheus

* vi /etc/systemd/system/prometheus.service

...

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

Group=prometheus

Type=simple

ExecStart=/usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.templates=/etc/prometheus/consoles --web.console.libraries=/etc/prometheus/console\_libraries

[Install]

WantedBy=multi-user.target

...

#Configuring Prometheus Configuration File

* cp /etc/prometheus/prometheus.yml /etc/prometheus/BKP\_prometheus.yml
* > /etc/prometheus/prometheus.yml
* vi /etc/prometheus/prometheus.yml

...

global:

scrape\_interval: 15s

evaluation\_interval: 15s

rule\_files:

# - "first\_rules.yml"

# - "second\_rules.yml"

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['10.10.100.171:9090']

- job\_name: 'spring-actuator'

metrics\_path: '/actuator/prometheus'

scrape\_interval: 5s

static\_configs:

- targets: ['10.10.100.171:2020']

basic\_auth:

username: 'username\_of\_endpoint'

password: 'password\_of\_endpoint'

...

#To validate the config

* /usr/local/bin/promtool check config /etc/prometheus/prometheus.yml

#Start The Prometheus Service

* systemctl start prometheus

#Enable the Prometheus service on system boot up

* systemctl enable prometheus

#Check the status of the Prometheus service

* systemctl status prometheus

#Prometheus service is listening on the port 9090

* netstat -nltp |grep prometheus

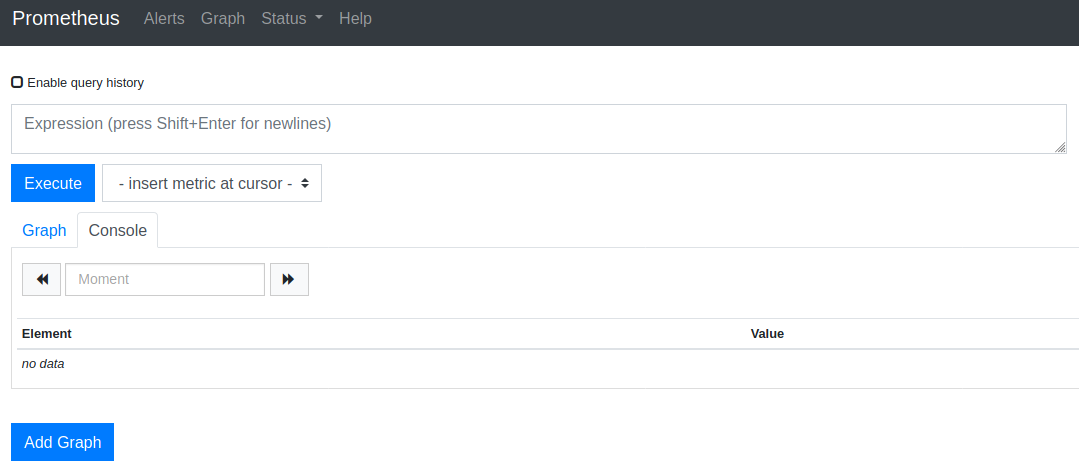
#Change firewall configuration to allow Prometheus port

* firewall-cmd --zone=public --add-port=9090/tcp –permanent
* firewall-cmd --reload

#To access the Prometheus Dashboard

http://<IP>:9090

#You will get the following page



#Under Status, Click Targets, we can see (below screenshot) all the targets monitored by the Prometheus

Graphical user interface, text, application

Description automatically generated

Prometheus installation and configuration done.

Verify Metrics from Spring Boot are showing up in Prometheus or not:

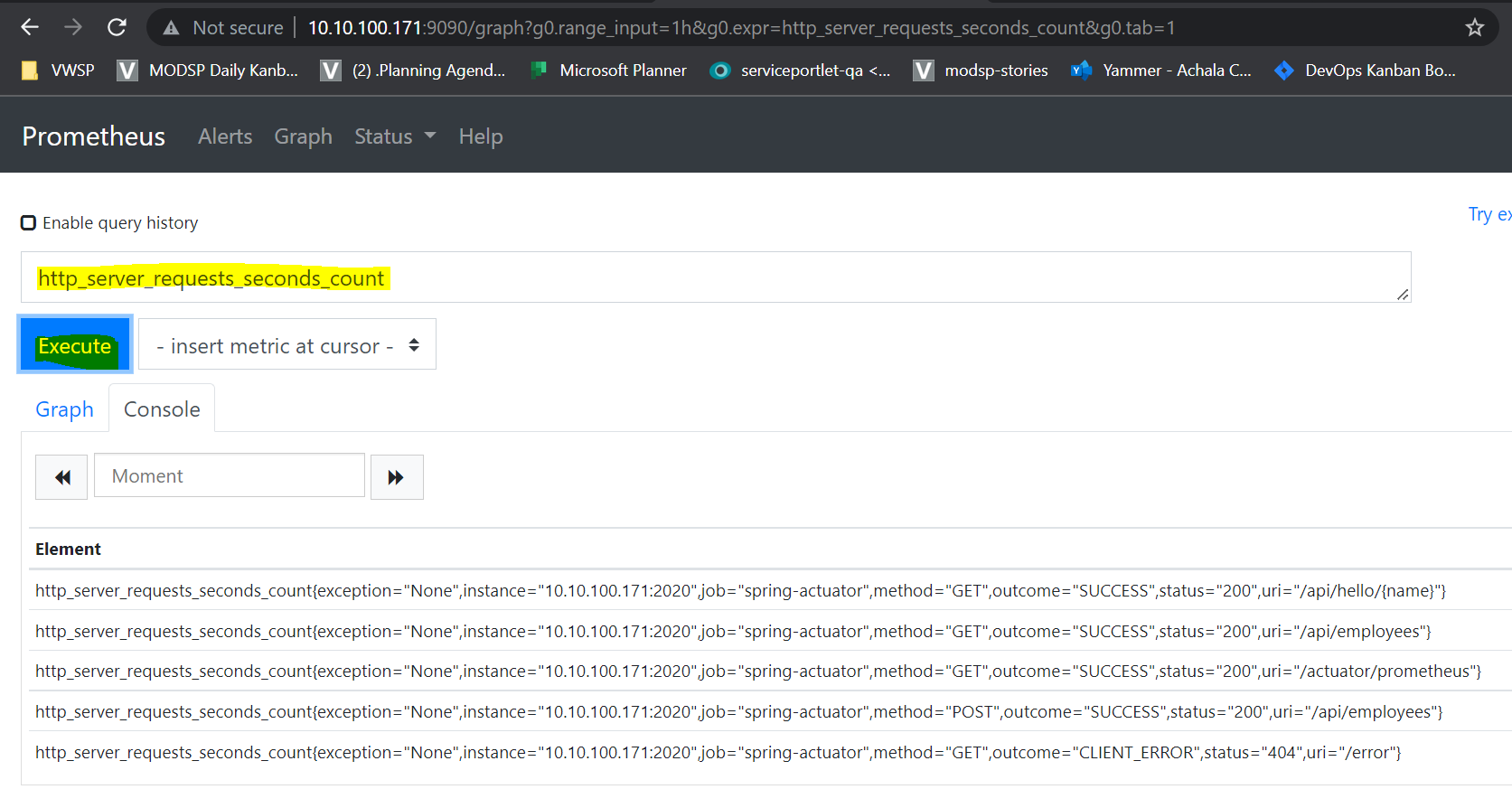
#Visit the following Spring Boot actuator end point and get anyone of the metric name

<http://10.10.100.171:2020/actuator/prometheus>

eg: one of the metrics from above url 🡪 *http\_server\_requests\_seconds\_count*

#Search whether the above metric is available in Prometheus dashboard or not

After navigating to Prometheus dashboard paste the metric name and click on Execute.



With the above screenshot we can conclude that Metrics from Spring Boot are showing up in Prometheus.

Next, we must install Grafana (to view more customized dashboard look and feel over Prometheus, we are installing Grafana)

**Grafana installation and configuration:**

#Check and update the sestatus

* sestatus
* setenforce 0

#Create a repo file and add the following content.

vi /etc/yum.repos.d/grafana.repo

...

[grafana]

name=grafana

baseurl=https://packages.grafana.com/oss/rpm

repo\_gpgcheck=1

enabled=1

gpgcheck=1

gpgkey=https://packages.grafana.com/gpg.key

sslverify=1

sslcacert=/etc/pki/tls/certs/ca-bundle.crt

...

#Install Grafana

sudo yum install grafana

#The Package does the following things:

…

Installs binary to /usr/sbin/grafana-server

Copies init.d script to /etc/init.d/grafana-server

Installs default file to /etc/sysconfig/grafana-server

Copies configuration file to /etc/grafana/grafana.ini

Installs systemd service (if systemd is available) name grafana-server.service

The default configuration uses a log file at /var/log/grafana/grafana.log

…

#Install additional font packages

yum install fontconfig

yum install freetype\*

yum install urw-fonts

#Enable Grafana Service

systemctl status grafana-server

systemctl start grafana-server

systemctl enable grafana-server

#Change firewall configuration to allow Grafana port

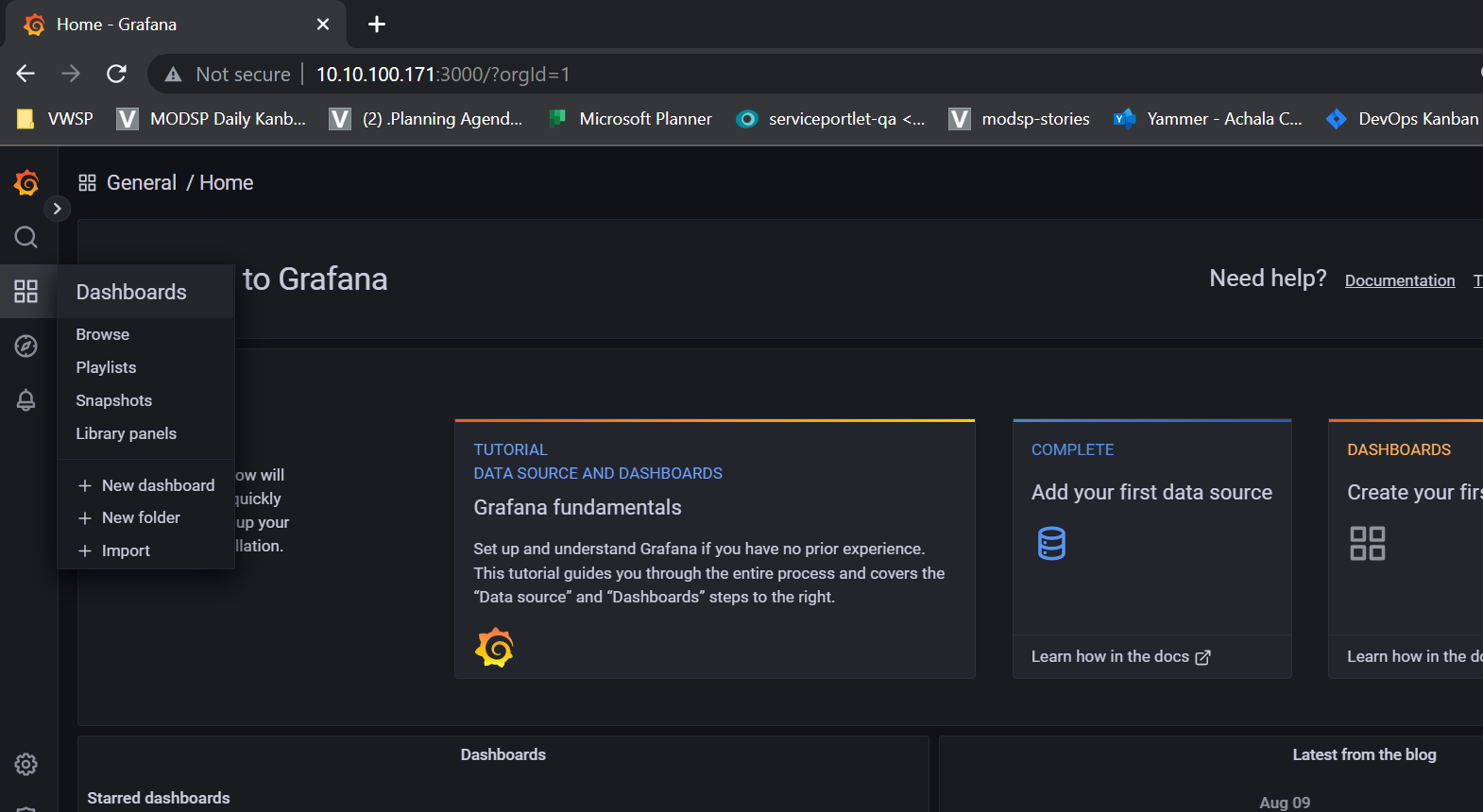
firewall-cmd --zone=public --add-port=3000/tcp --permanent

firewall-cmd --reload

#Use the following URL to access the Grafana web interface

http://<IP>:3000/

Enter “admin” in the login and password fields for first-time use; then it will ask you to change the password.

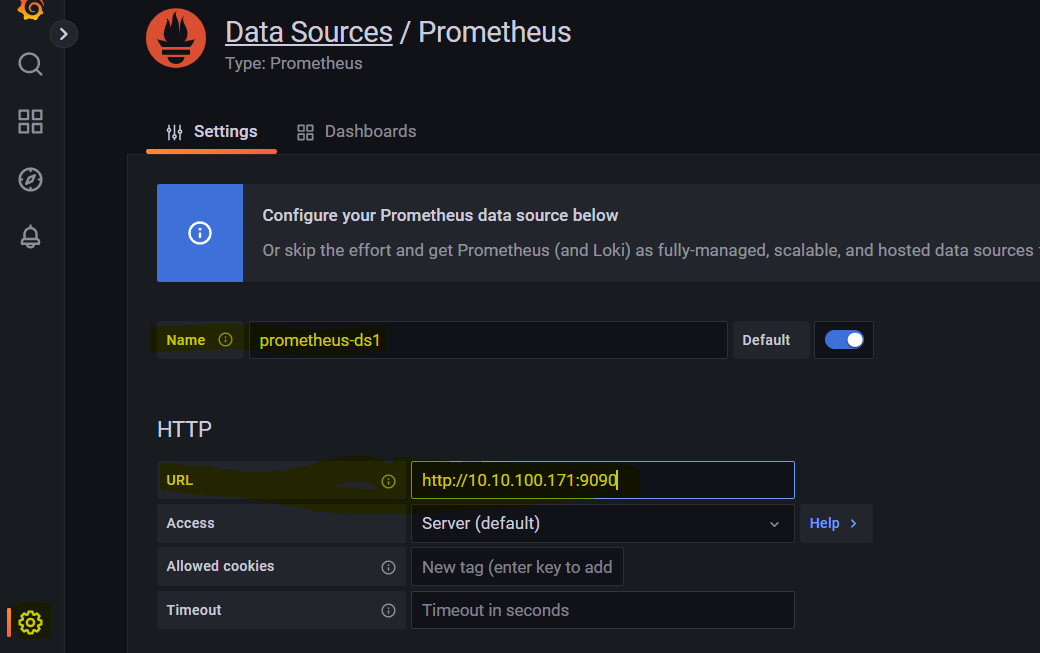


In the navigation bar on the left, when you click on the gear icon, a submenu named datasource is displayed, click on it

(Go to configuration > Add data source > prometheus)

#Add our previously configured Prometheus as the datasource.

provide prometheus url : <http://10.10.100.171:9090>



remaining configuration as it is. Click on “save & test” (at the end)

Graphical user interface, website

Description automatically generated

If everything is fine, then it will show “Data source is working” when you click on “save & test” button.

With this, we have successfully added our data source (Prometheus) in Grafana.

Now, we can create our own dashboards, or we can import (json file) the existing built-in dashboard.

Here, I have downloaded built-in dashboard json file from the following url:

<https://grafana.com/grafana/dashboards/12900>

Download the file from above url and import into Grafana as a monitoring dashboard for springboot application.

Navigate to Dashboard > import

A screenshot of a computer

Description automatically generated with medium confidence

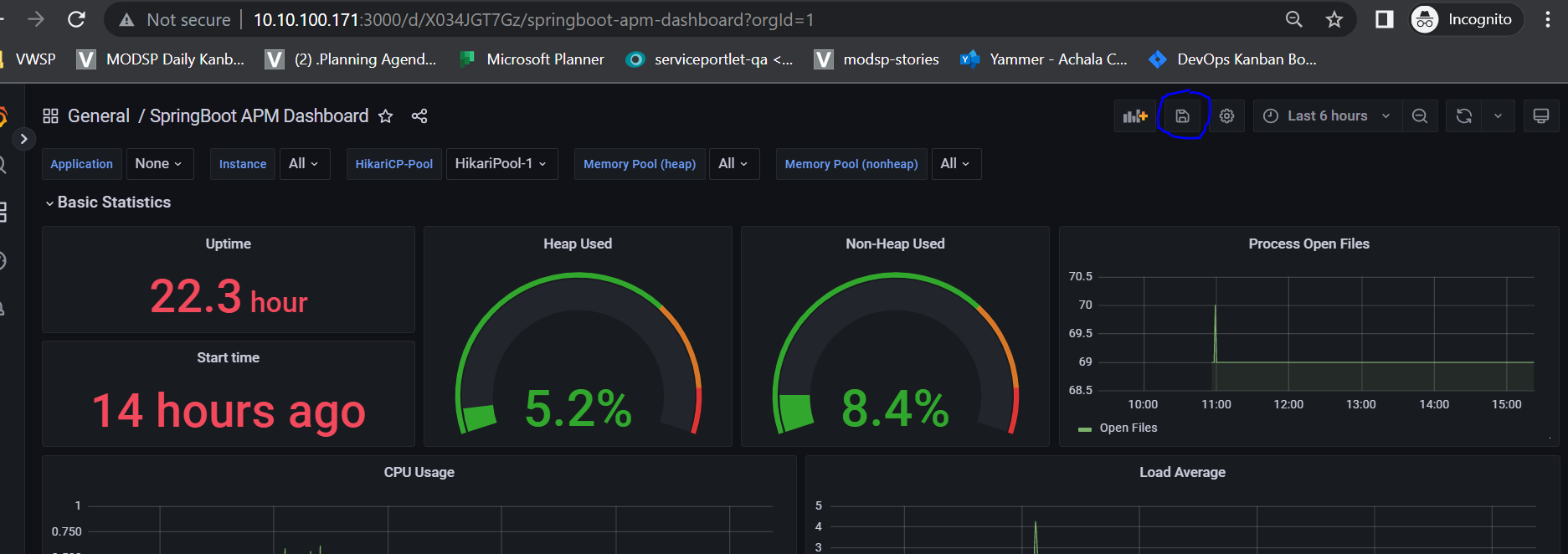
Click on “Upload JSON file” and select the json file from your filesystem.

Graphical user interface, text, application

Description automatically generated

Select the data source name (as shown in the screenshot) and then click on “Import” button.

Once you click on “Import” button it will prompt you to the dashboard.



Save the dashboard (in screenshot, rounded with blue color).

Thank you!

Achala Devops.