Observabilidade com SpringBoot Actuator + Prometheus + Grafana

Projeto SpringBoot

- Crie projeto SpringBoot com as seguintes dependências:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>

<dependency>
    <groupId>io.micrometer</groupId>
    <artifactId>micrometer-registry-prometheus</artifactId>
    <version>1.12.5</version>
</dependency>
```

- Crie um controller para testarmos as métricas de endpoint:

```
@RestController
public class TestController {

    @GetMapping(path = "/hello")
    public ResponseEntity<String> hello (@PathParam("name") String name) {
        if("error".equalsIgnoreCase(name)) {
            return new ResponseEntity<>(("ERRO INESPERADO"), HttpStatus.INTERNAL_SERVER_ERROR);
        }
        return ResponseEntity.ok("Hello, " + name);
    }

}
```

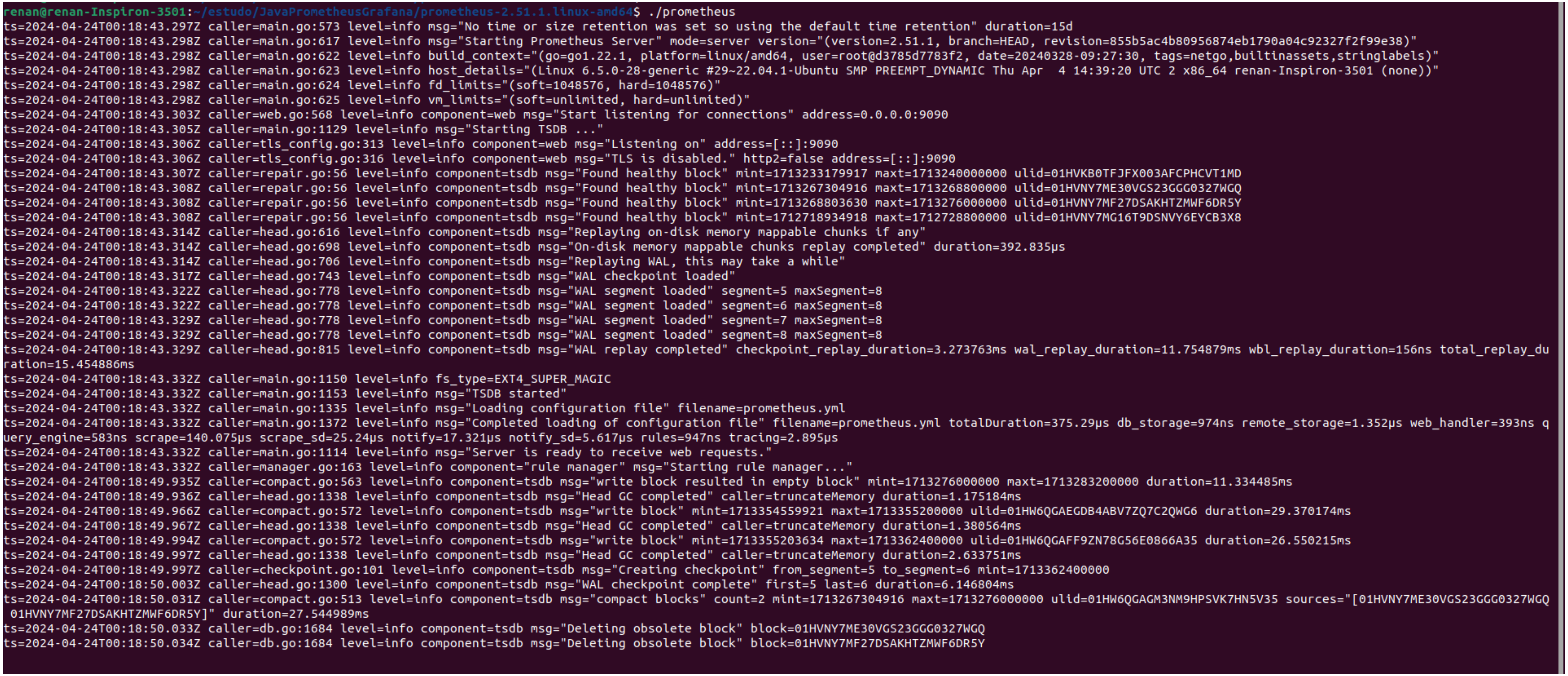
- No arquivo de configuração do projeto (application.yml)

```
management:
  endpoints:
    enabled-by-default: false
    web:
      exposure:
        include: 'health, prometheus'
  endpoint:
    health:
      enabled: true
      show-details: always
    prometheus:
      enabled: true
```

Prometheus

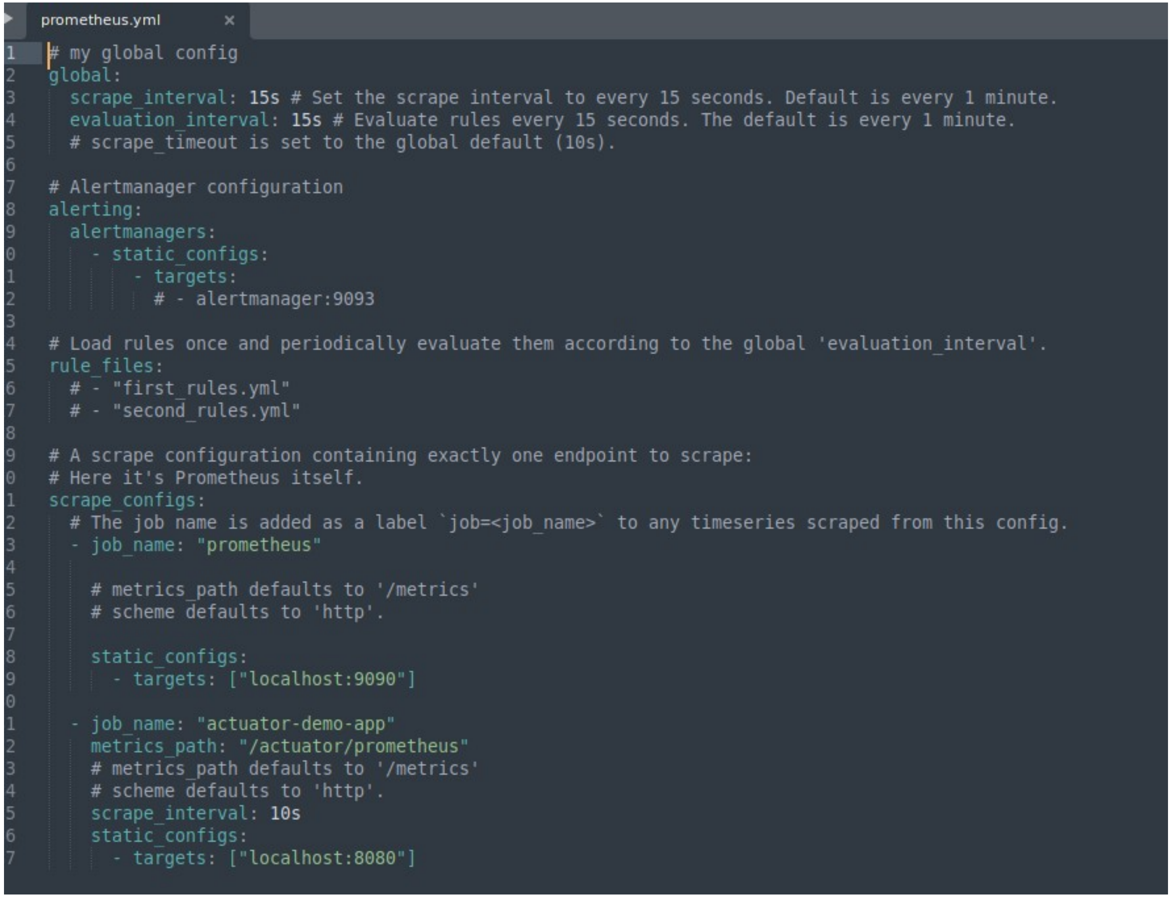
- Prometheus é o coletor de métricas que irá pegar da aplicação e fornecer ao Grafana para serem exibidas.
- Download: https://prometheus.io/download/

- Depois de extraído execute ./prometheus dentro da pasta.



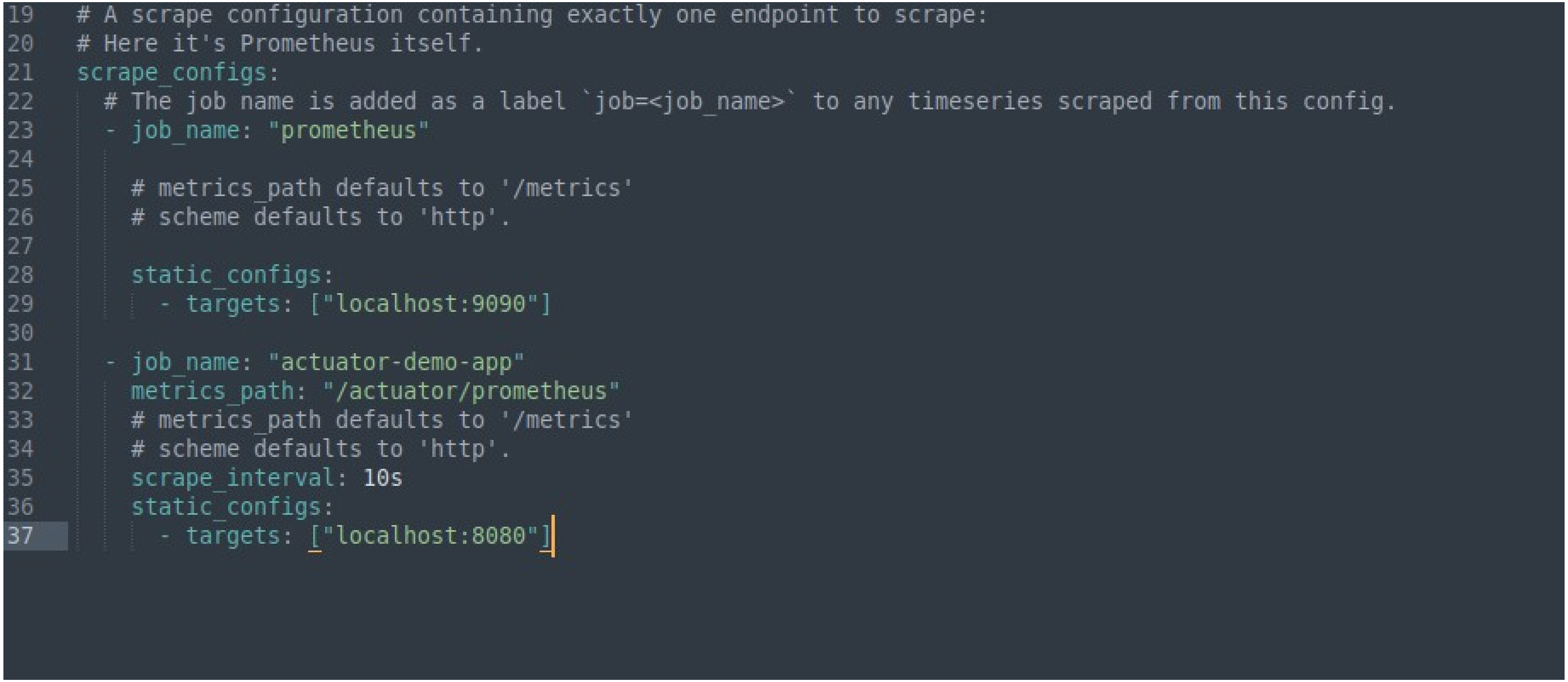
```
renan@renan-Inspiron-3501:~/estudo/JavaPrometheusGrafana/prometheus-2.51.1.linux-amd64$ ./prometheus
ts=2024-04-24T00:18:43.297Z caller=main.go:573 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2024-04-24T00:18:43.298Z caller=main.go:617 level=info msg="Starting Prometheus Server" mode=server version="(version=2.51.1, branch=HEAD, revision=855b5ac4b80956874eb1790a04c92327f2f99e38)"
ts=2024-04-24T00:18:43.298Z caller=main.go:622 level=info build_context="(go=go1.22.1, platform=linux/amd64, user=root@d3785d7783f2, date=20240328-09:27:30, tags=netgo,builtinassets,stringlabels)"
ts=2024-04-24T00:18:43.298Z caller=main.go:623 level=info host_details="(Linux 6.5.0-28-generic #29~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Apr  4 14:39:20 UTC 2 x86_64 renan-Inspiron-3501 (none))"
ts=2024-04-24T00:18:43.298Z caller=main.go:624 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2024-04-24T00:18:43.298Z caller=main.go:625 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-04-24T00:18:43.303Z caller=web.go:568 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2024-04-24T00:18:43.305Z caller=main.go:1129 level=info msg="Starting TSDB ..."
ts=2024-04-24T00:18:43.306Z caller=tls_config.go:313 level=info component=web msg="Listening on" address=[::]:9090
ts=2024-04-24T00:18:43.306Z caller=tls_config.go:316 level=info component=web msg="TLS is disabled." http2=false address=[::]:9090
ts=2024-04-24T00:18:43.307Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1713233179917 maxt=1713240000000 ulid=01HVKB0TFJFX003AFCPHCVT1MD
ts=2024-04-24T00:18:43.308Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1713267304916 maxt=1713268800000 ulid=01HVNY7ME30VGS23GGG0327WGQ
ts=2024-04-24T00:18:43.308Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1713268803630 maxt=1713276000000 ulid=01HVNY7MF27DSAKHTZMWF6DR5Y
ts=2024-04-24T00:18:43.308Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1712718934918 maxt=1712728800000 ulid=01HVNY7MG16T9DSNVY6EYCB3X8
ts=2024-04-24T00:18:43.314Z caller=head.go:616 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2024-04-24T00:18:43.314Z caller=head.go:698 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=392.835µs
ts=2024-04-24T00:18:43.314Z caller=head.go:706 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-04-24T00:18:43.317Z caller=head.go:743 level=info component=tsdb msg="WAL checkpoint loaded"
ts=2024-04-24T00:18:43.322Z caller=head.go:778 level=info component=tsdb msg="WAL segment loaded" segment=5 maxSegment=8
ts=2024-04-24T00:18:43.322Z caller=head.go:778 level=info component=tsdb msg="WAL segment loaded" segment=6 maxSegment=8
ts=2024-04-24T00:18:43.329Z caller=head.go:778 level=info component=tsdb msg="WAL segment loaded" segment=7 maxSegment=8
ts=2024-04-24T00:18:43.329Z caller=head.go:778 level=info component=tsdb msg="WAL segment loaded" segment=8 maxSegment=8
ts=2024-04-24T00:18:43.329Z caller=head.go:815 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=3.273763ms wal_replay_duration=11.754879ms wbl_replay_duration=156ns total_replay_du
ration=15.454886ms
ts=2024-04-24T00:18:43.332Z caller=main.go:1150 level=info fs_type=EXT4_SUPER_MAGIC
ts=2024-04-24T00:18:43.332Z caller=main.go:1153 level=info msg="TSDB started"
ts=2024-04-24T00:18:43.332Z caller=main.go:1335 level=info msg="Loading configuration file" filename=prometheus.yml
ts=2024-04-24T00:18:43.332Z caller=main.go:1372 level=info msg="Completed loading of configuration file" filename=prometheus.yml totalDuration=375.29µs db_storage=974ns remote_storage=1.352µs web_handler=393ns q
uery_engine=583ns scrape=140.075µs scrape_sd=25.24µs notify=17.321µs notify_sd=5.617µs rules=947ns tracing=2.895µs
ts=2024-04-24T00:18:43.332Z caller=main.go:1114 level=info msg="Server is ready to receive web requests."
ts=2024-04-24T00:18:43.332Z caller=manager.go:163 level=info component="rule manager" msg="Starting rule manager..."
ts=2024-04-24T00:18:49.935Z caller=compact.go:563 level=info component=tsdb msg="write block resulted in empty block" mint=1713276000000 maxt=1713283200000 duration=11.334485ms
ts=2024-04-24T00:18:49.936Z caller=head.go:1338 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=1.175184ms
ts=2024-04-24T00:18:49.966Z caller=compact.go:572 level=info component=tsdb msg="write block" mint=1713354559921 maxt=1713355200000 ulid=01HW6QGAEGDB4ABV7ZQ7C2QWG6 duration=29.370174ms
ts=2024-04-24T00:18:49.967Z caller=head.go:1338 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=1.380564ms
ts=2024-04-24T00:18:49.994Z caller=compact.go:572 level=info component=tsdb msg="write block" mint=1713355203634 maxt=1713362400000 ulid=01HW6QGAFF9ZN78G56E0866A35 duration=26.550215ms
ts=2024-04-24T00:18:49.997Z caller=head.go:1338 level=info component=tsdb msg="Head GC completed" caller=truncateMemory duration=2.633751ms
ts=2024-04-24T00:18:49.997Z caller=checkpoint.go:101 level=info component=tsdb msg="Creating checkpoint" from_segment=5 to_segment=6 mint=1713362400000
ts=2024-04-24T00:18:50.003Z caller=head.go:1300 level=info component=tsdb msg="WAL checkpoint complete" first=5 last=6 duration=6.146804ms
ts=2024-04-24T00:18:50.031Z caller=compact.go:513 level=info component=tsdb msg="compact blocks" count=2 mint=1713267304916 maxt=1713276000000 ulid=01HW6QGAGM3NM9HPSVK7HN5V35 sources="[01HVNY7ME30VGS23GGG0327WGQ
 01HVNY7MF27DSAKHTZMWF6DR5Y]" duration=27.544989ms
ts=2024-04-24T00:18:50.033Z caller=db.go:1684 level=info component=tsdb msg="Deleting obsolete block" block=01HVNY7ME30VGS23GGG0327WGQ
ts=2024-04-24T00:18:50.034Z caller=db.go:1684 level=info component=tsdb msg="Deleting obsolete block" block=01HVNY7MF27DSAKHTZMWF6DR5Y
```

- O arquivo de configuração do prometheus é o prometheus.yml, ali você pode fazer as alterações que desejar.



```yaml
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "actuator-demo-app"
    metrics_path: "/actuator/prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    scrape_interval: 10s
    static_configs:
      - targets: ["localhost:8080"]
```

- Veja que configuramos o nome do coletor, a porta e endpoint do projeto que fornecerá as métricas:

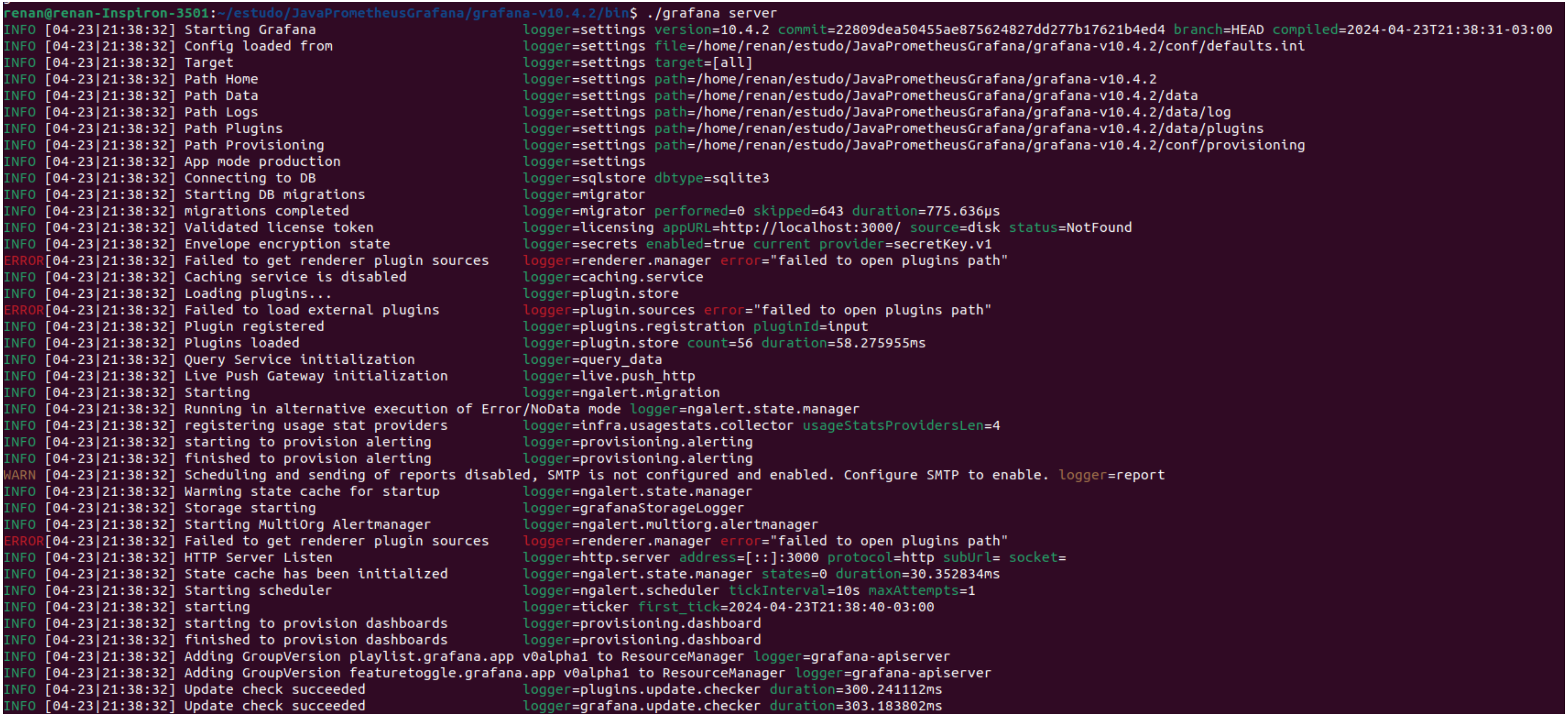


```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "actuator-demo-app"
    metrics_path: "/actuator/prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    scrape_interval: 10s
    static_configs:
      - targets: ["localhost:8080"]
```

Grafana

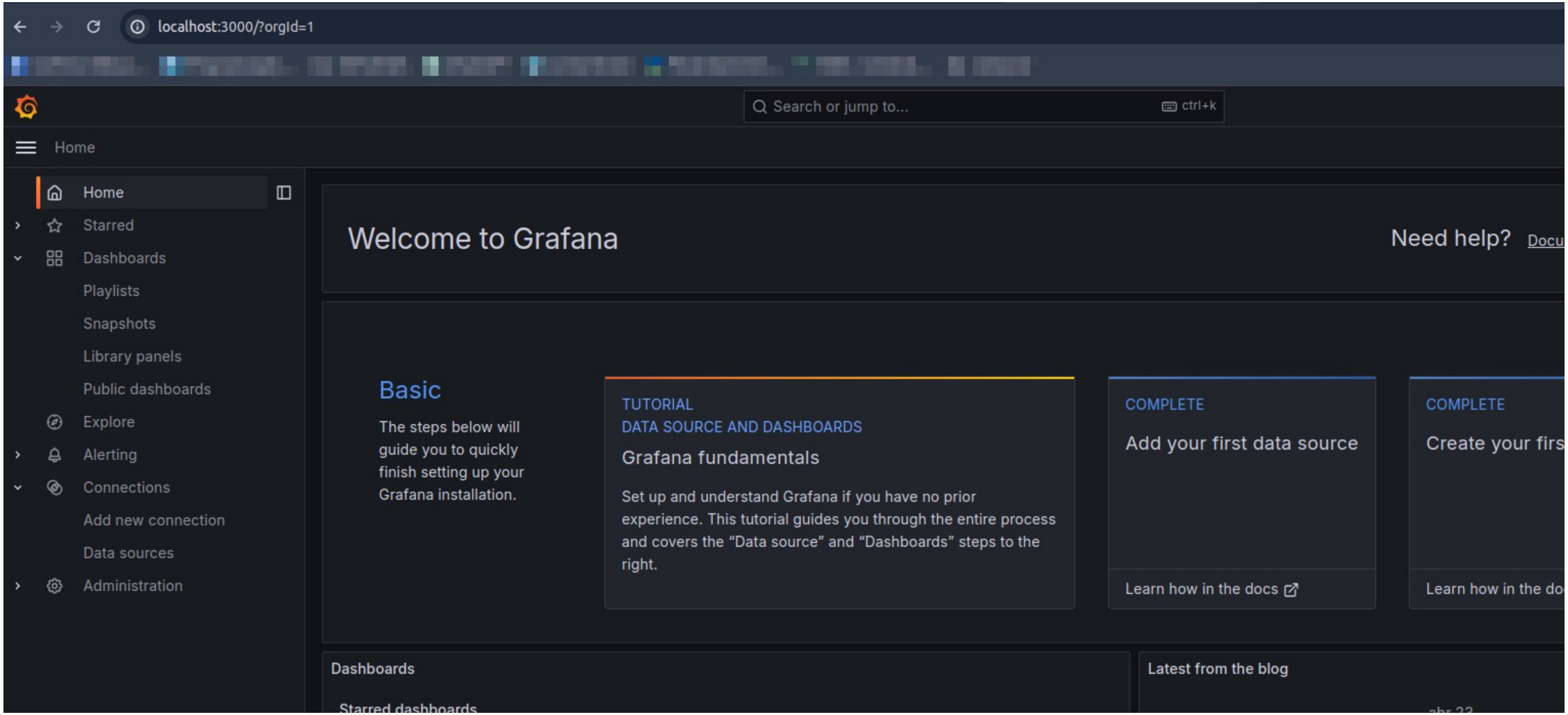
- Grafana é um software de exibição de métricas via dashboards, onde podemos configurar e exibir as métricas coletadas pelo Prometheus.
- Download: https://grafana.com/grafana/download

- Após baixar e extrair, execute dentro da pasta /bin o comando: ./grafana server



```
renan@renan-Inspiron-3501:~/estudo/JavaPrometheusGrafana/grafana-v10.4.2/bin$ ./grafana server
INFO [04-23|21:38:32] Starting Grafana                         logger=settings version=10.4.2 commit=22809dea50455ae875624827dd277b17621b4ed4 branch=HEAD compiled=2024-04-23T21:38:31-03:00
INFO [04-23|21:38:32] Config loaded from                       logger=settings file=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2/conf/defaults.ini
INFO [04-23|21:38:32] Target                                   logger=settings target=[all]
INFO [04-23|21:38:32] Path Home                                logger=settings path=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2
INFO [04-23|21:38:32] Path Data                                logger=settings path=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2/data
INFO [04-23|21:38:32] Path Logs                                logger=settings path=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2/data/log
INFO [04-23|21:38:32] Path Plugins                             logger=settings path=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2/data/plugins
INFO [04-23|21:38:32] Path Provisioning                        logger=settings path=/home/renan/estudo/JavaPrometheusGrafana/grafana-v10.4.2/conf/provisioning
INFO [04-23|21:38:32] App mode production                      logger=settings
INFO [04-23|21:38:32] Connecting to DB                         logger=sqlstore dbtype=sqlite3
INFO [04-23|21:38:32] Starting DB migrations                   logger=migrator
INFO [04-23|21:38:32] migrations completed                     logger=migrator performed=0 skipped=643 duration=775.636µs
INFO [04-23|21:38:32] Validated license token                  logger=licensing appURL=http://localhost:3000/ source=disk status=NotFound
INFO [04-23|21:38:32] Envelope encryption state                logger=secrets enabled=true current provider=secretKey.v1
ERROR[04-23|21:38:32] Failed to get renderer plugin sources    logger=renderer.manager error="failed to open plugins path"
INFO [04-23|21:38:32] Caching service is disabled              logger=caching.service
INFO [04-23|21:38:32] Loading plugins...                       logger=plugin.store
ERROR[04-23|21:38:32] Failed to load external plugins          logger=plugin.sources error="failed to open plugins path"
INFO [04-23|21:38:32] Plugin registered                        logger=plugins.registration pluginId=input
INFO [04-23|21:38:32] Plugins loaded                           logger=plugin.store count=56 duration=58.275955ms
INFO [04-23|21:38:32] Query Service initialization             logger=query_data
INFO [04-23|21:38:32] Live Push Gateway initialization         logger=live.push_http
INFO [04-23|21:38:32] Starting                                 logger=ngalert.migration
INFO [04-23|21:38:32] Running in alternative execution of Error/NoData mode logger=ngalert.state.manager
INFO [04-23|21:38:32] registering usage stat providers         logger=infra.usagestats.collector usageStatsProvidersLen=4
INFO [04-23|21:38:32] starting to provision alerting           logger=provisioning.alerting
INFO [04-23|21:38:32] finished to provision alerting           logger=provisioning.alerting
WARN [04-23|21:38:32] Scheduling and sending of reports disabled, SMTP is not configured and enabled. Configure SMTP to enable. logger=report
INFO [04-23|21:38:32] Warming state cache for startup          logger=ngalert.state.manager
INFO [04-23|21:38:32] Storage starting                         logger=grafanaStorageLogger
INFO [04-23|21:38:32] Starting MultiOrg Alertmanager           logger=ngalert.multiorg.alertmanager
ERROR[04-23|21:38:32] Failed to get renderer plugin sources    logger=renderer.manager error="failed to open plugins path"
INFO [04-23|21:38:32] HTTP Server Listen                       logger=http.server address=[::]:3000 protocol=http subUrl= socket=
INFO [04-23|21:38:32] State cache has been initialized         logger=ngalert.state.manager states=0 duration=30.352834ms
INFO [04-23|21:38:32] Starting scheduler                       logger=ngalert.scheduler tickInterval=10s maxAttempts=1
INFO [04-23|21:38:32] starting                                 logger=ticker first_tick=2024-04-23T21:38:40-03:00
INFO [04-23|21:38:32] starting to provision dashboards         logger=provisioning.dashboard
INFO [04-23|21:38:32] finished to provision dashboards         logger=provisioning.dashboard
INFO [04-23|21:38:32] Adding GroupVersion playlist.grafana.app v0alpha1 to ResourceManager logger=grafana-apiserver
INFO [04-23|21:38:32] Adding GroupVersion featuretoggle.grafana.app v0alpha1 to ResourceManager logger=grafana-apiserver
INFO [04-23|21:38:32] Update check succeeded                   logger=plugins.update.checker duration=300.241112ms
INFO [04-23|21:38:32] Update check succeeded                   logger=grafana.update.checker duration=303.183802ms
```

- Ele estará disponível via browser no endereço: http://localhost:3000/



Welcome to Grafana

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS

Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the “Data source” and “Dashboards” steps to the right.

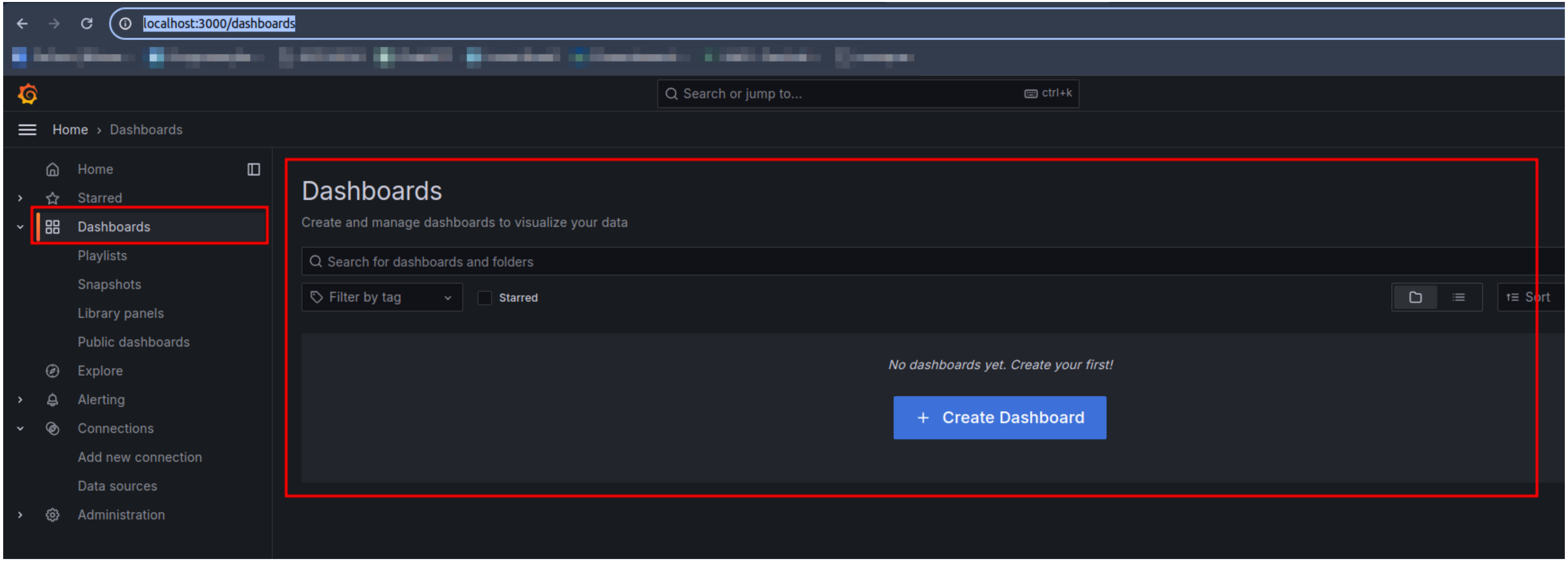
COMPLETE

Add your first data source

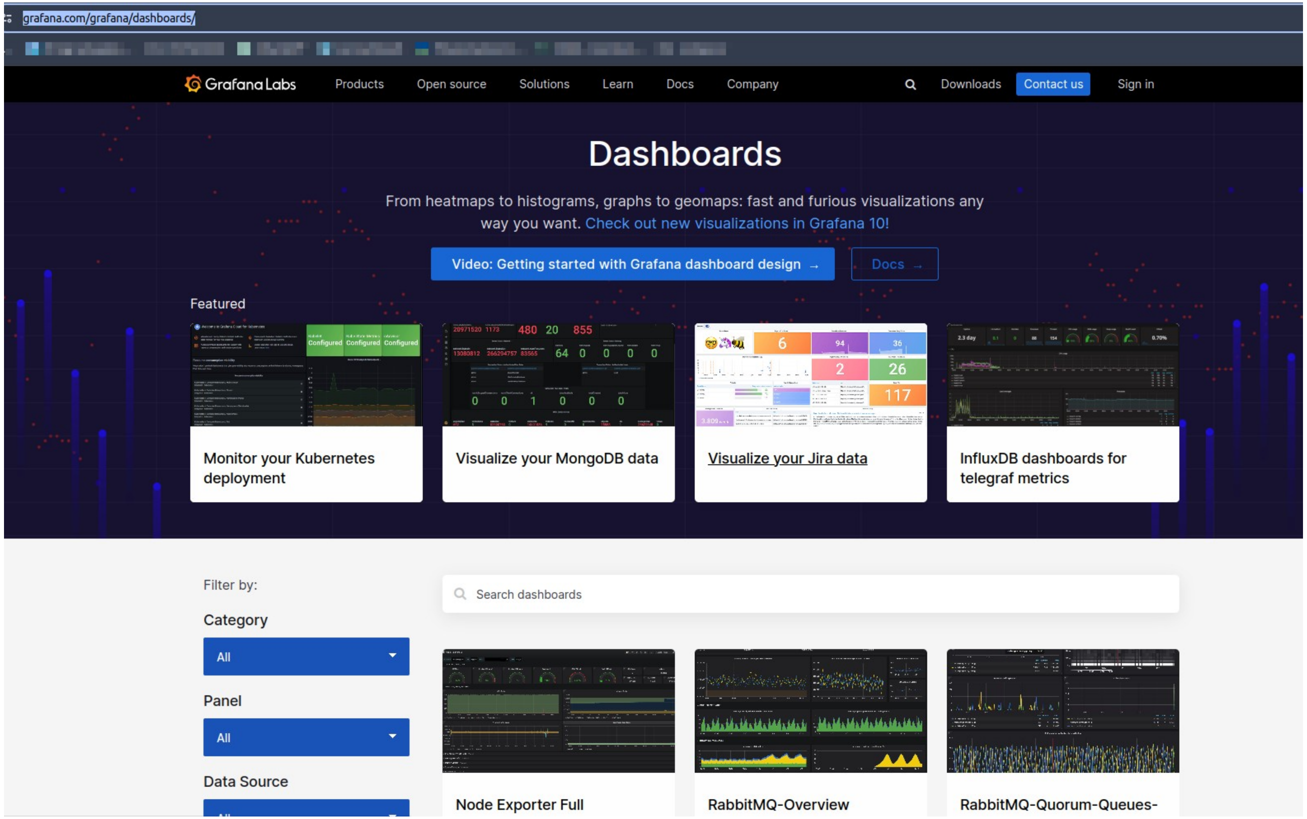
COMPLETE

Create your firs

- Em Dashboards no menu esquerdo é onde veremos e criaremos nossos quadros:



- Há vários modelos prontos em: https://grafana.com/grafana/dashboards/



Dashboards

From heatmaps to histograms, graphs to geomaps: fast and furious visualizations any way you want. Check out new visualizations in Grafana 10!

Featured

Monitor your Kubernetes deployment

Visualize your MongoDB data

Visualize your Jira data

InfluxDB dashboards for telegraf metrics

Filter by:

Category: All

Panel: All

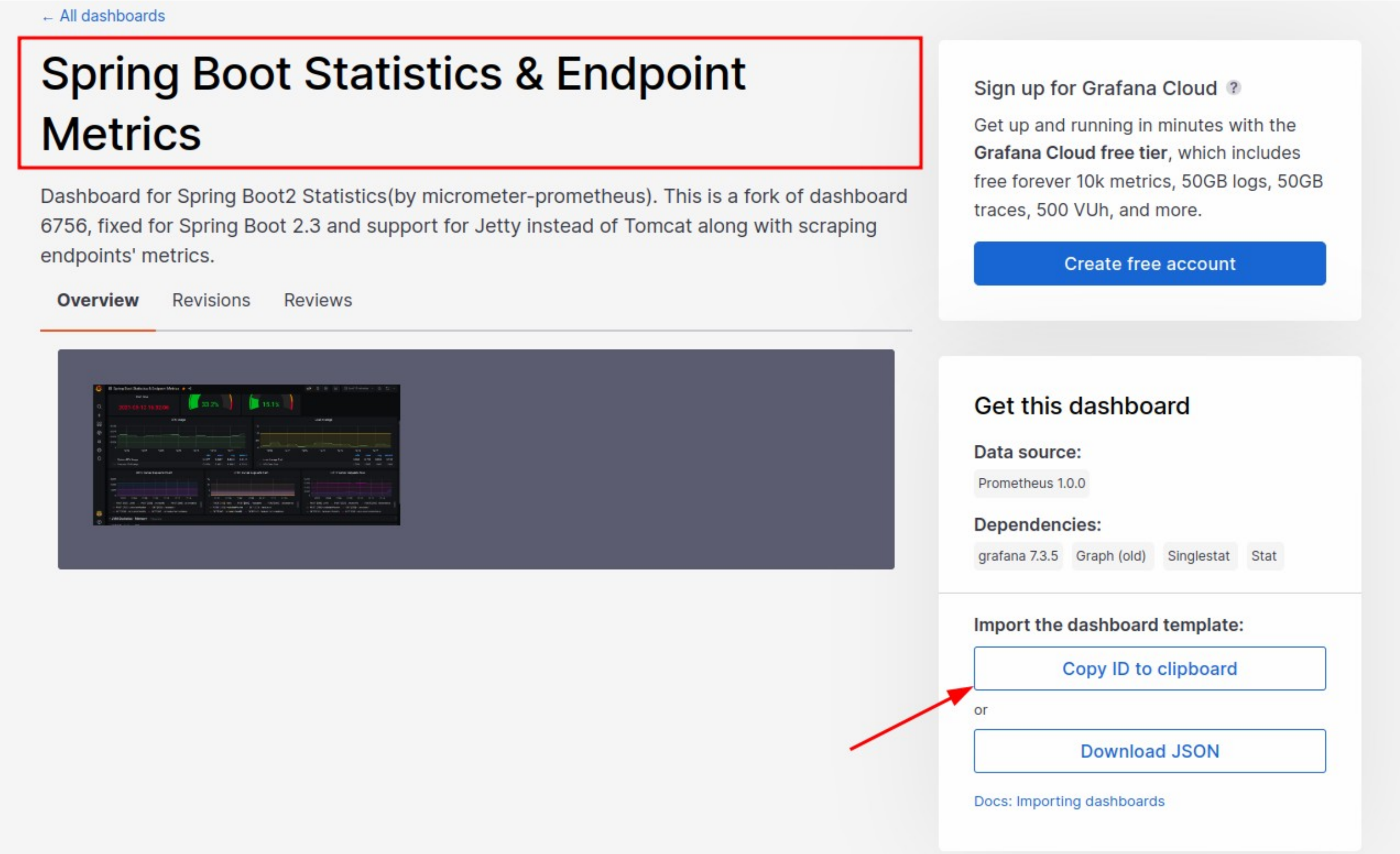
Data Source

Node Exporter Full

RabbitMQ-Overview

RabbitMQ-Quorum-Queues-

- Para esse exemplo vamos usar seguinte:



← All dashboards

Spring Boot Statistics & Endpoint Metrics

Dashboard for Spring Boot2 Statistics(by micrometer-prometheus). This is a fork of dashboard 6756, fixed for Spring Boot 2.3 and support for Jetty instead of Tomcat along with scraping endpoints' metrics.

Overview Revisions Reviews

Sign up for Grafana Cloud ?

Get up and running in minutes with the **Grafana Cloud free tier**, which includes free forever 10k metrics, 50GB logs, 50GB traces, 500 VUh, and more.

Create free account

Get this dashboard

Data source:

Prometheus 1.0.0

Dependencies:

grafana 7.3.5 Graph (old) Singlestat Stat

Import the dashboard template:

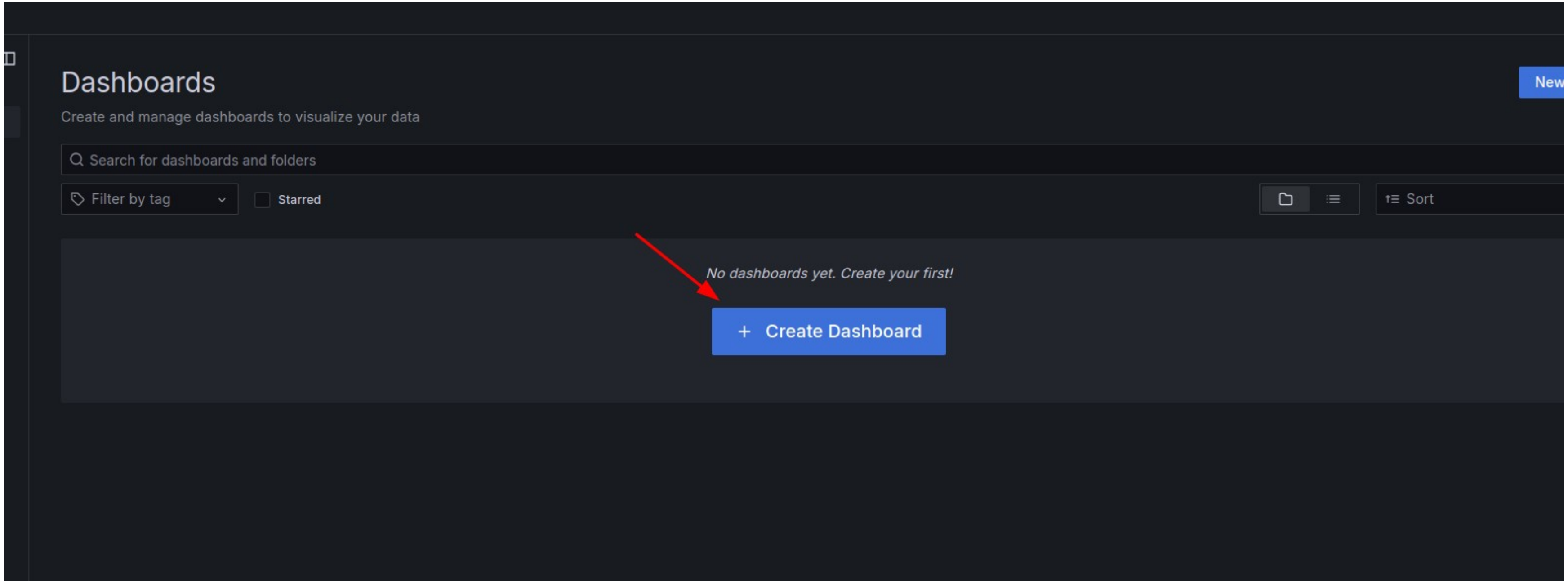
Copy ID to clipboard

or

Download JSON

Docs: Importing dashboards

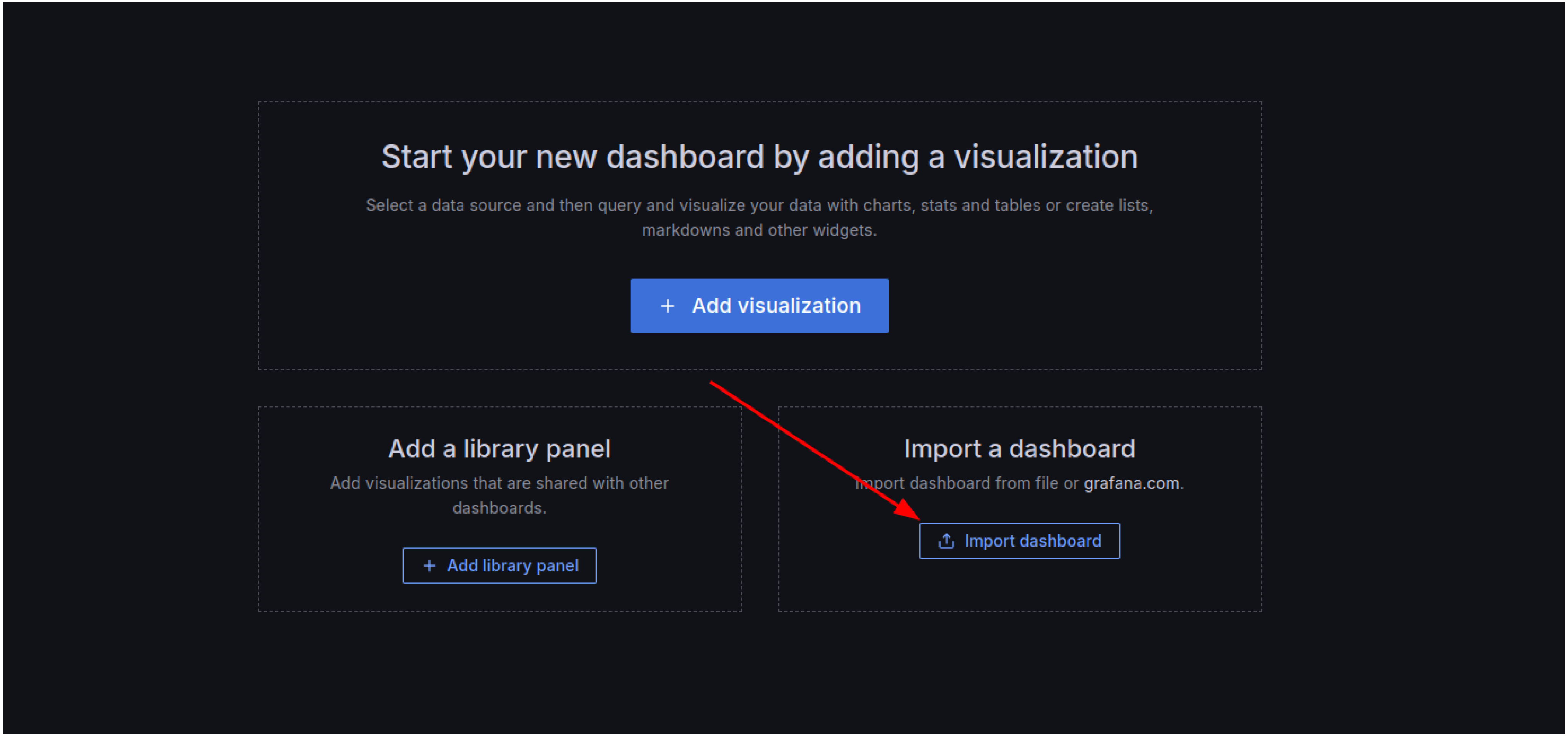
- Voltando no Grafana:



Dashboards

Create and manage dashboards to visualize your data

No dashboards yet. Create your first!



Start your new dashboard by adding a visualization

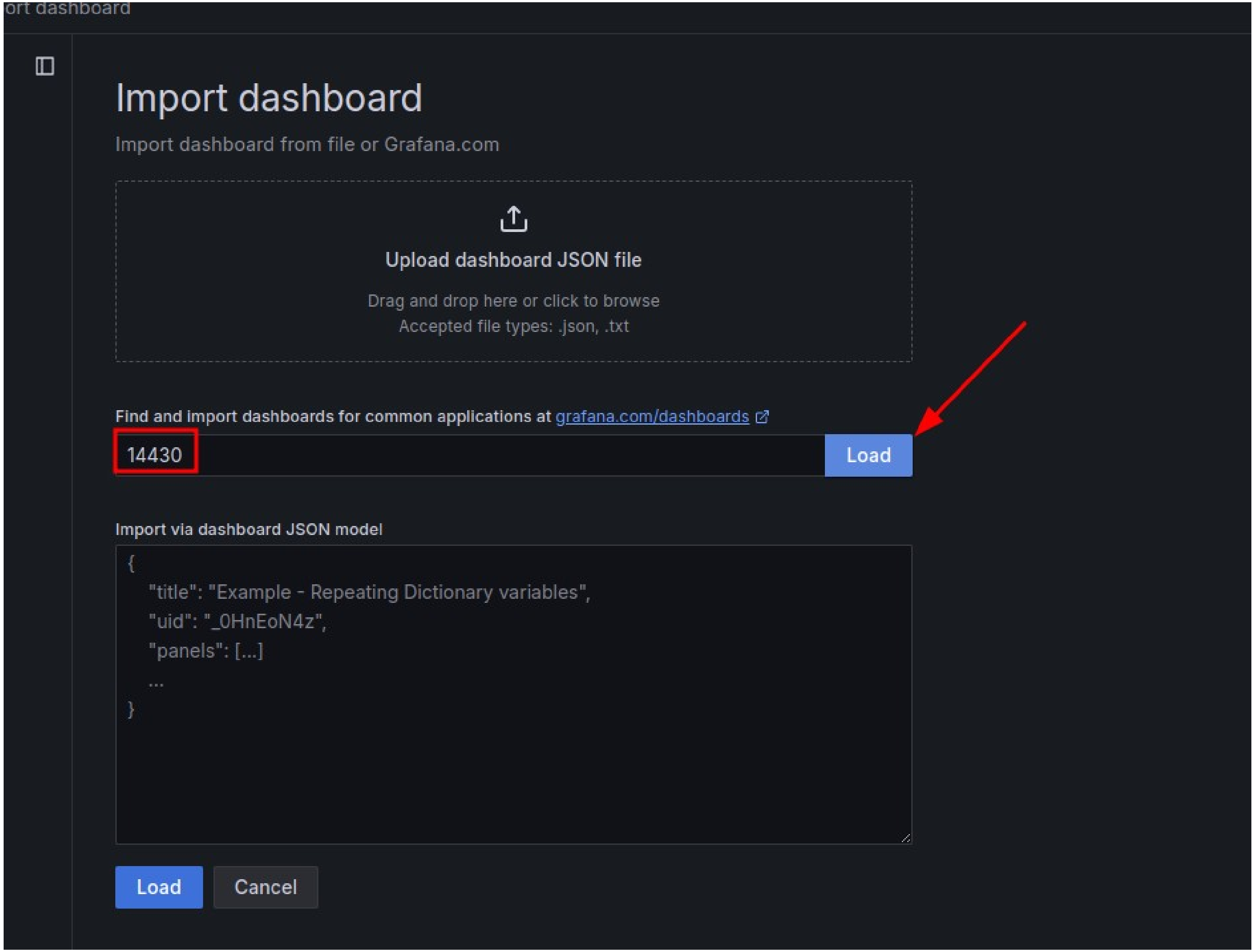
Select a data source and then query and visualize your data with charts, stats and tables or create lists, markdowns and other widgets.

Add a library panel

Add visualizations that are shared with other dashboards.

Import a dashboard

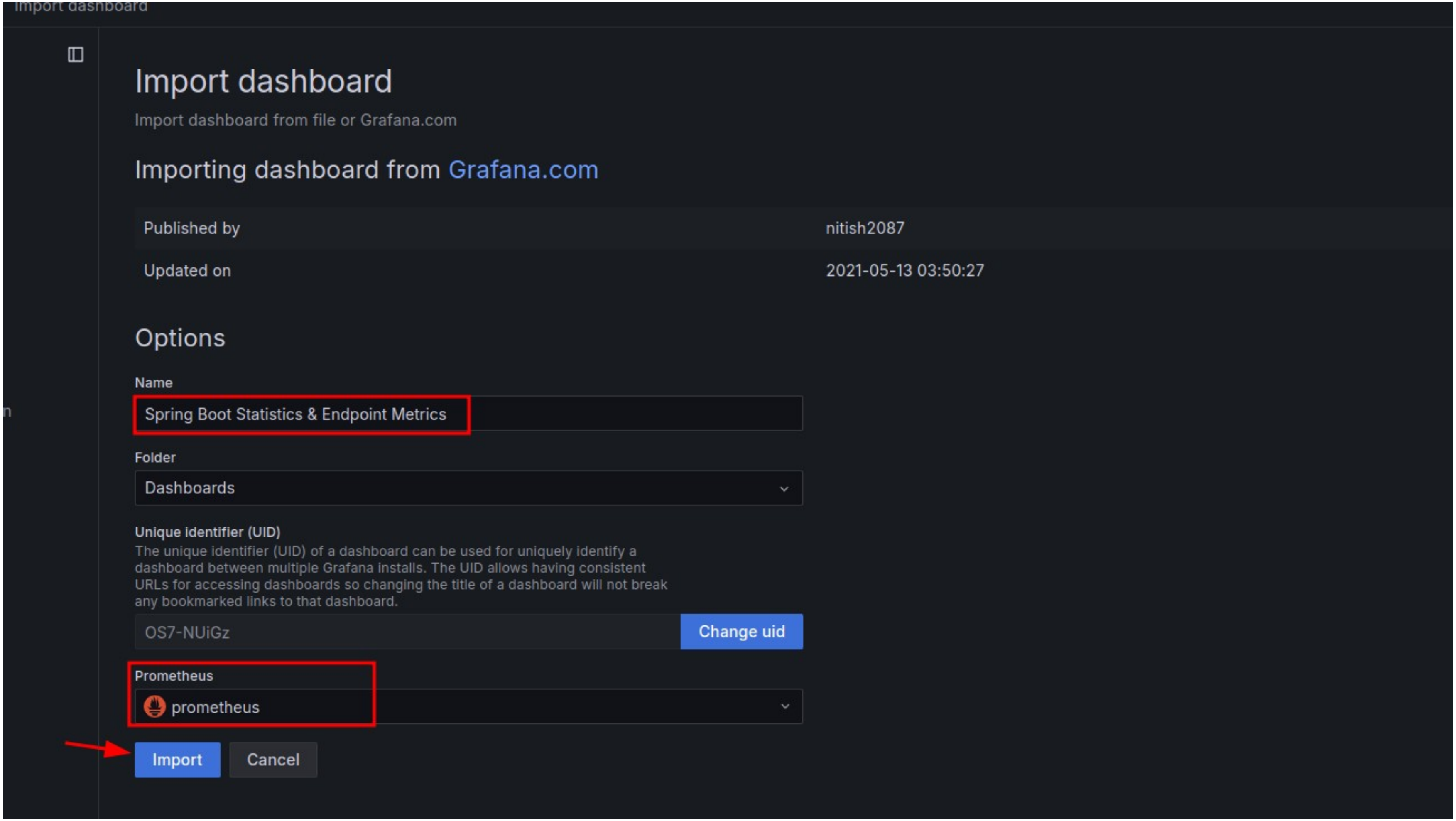
Import dashboard from file or grafana.com.



ort dashboard
Import dashboard
Import dashboard from file or Grafana.com
Upload dashboard JSON file
Drag and drop here or click to browse
Accepted file types: .json, .txt
Find and import dashboards for common applications at grafana.com/dashboards
Load
Import via dashboard JSON model
Load
Cancel

14430

```
{
    "title": "Example - Repeating Dictionary variables",
    "uid": "_0HnEoN4z",
    "panels": [...]
    ...
}
```



Import dashboard

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by	nitish2087
Updated on	2021-05-13 03:50:27

Options

Name: Spring Boot Statistics & Endpoint Metrics

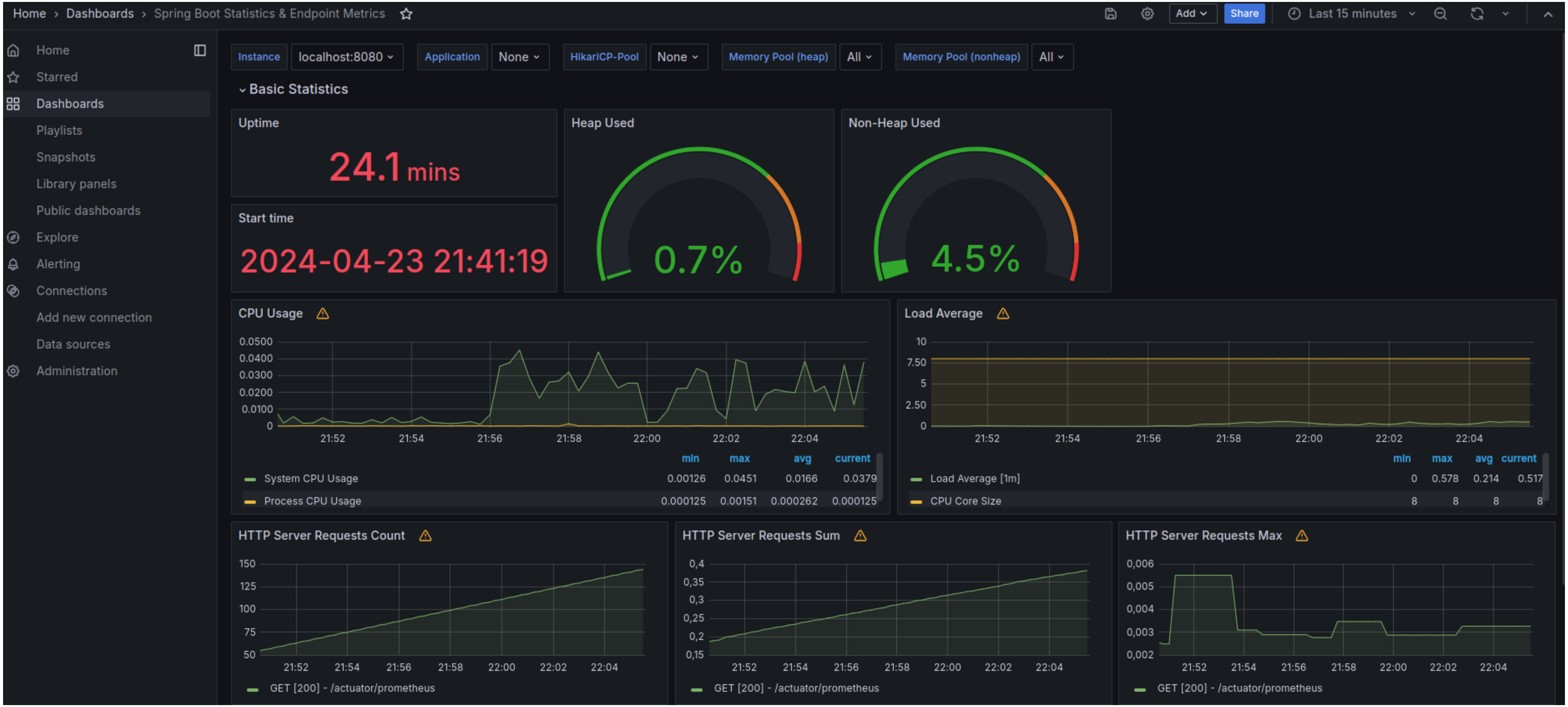
Folder: Dashboards

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

OS7-NUiGz

Prometheus: prometheus



Instance: localhost:8080 | Application: None | HikariCP-Pool: None | Memory Pool (heap): All | Memory Pool (nonheap): All

Basic Statistics

Uptime: 24.1 mins

Start time: 2024-04-23 21:41:19

Heap Used: 0.7%

Non-Heap Used: 4.5%

CPU Usage

	min	max	avg	current
System CPU Usage	0.00126	0.0451	0.0166	0.0379
Process CPU Usage	0.000125	0.00151	0.000262	0.000125

Load Average

	min	max	avg	current
Load Average [1m]	0	0.578	0.214	0.517
CPU Core Size	8	8	8	8

HTTP Server Requests Count

GET [200] - /actuator/prometheus

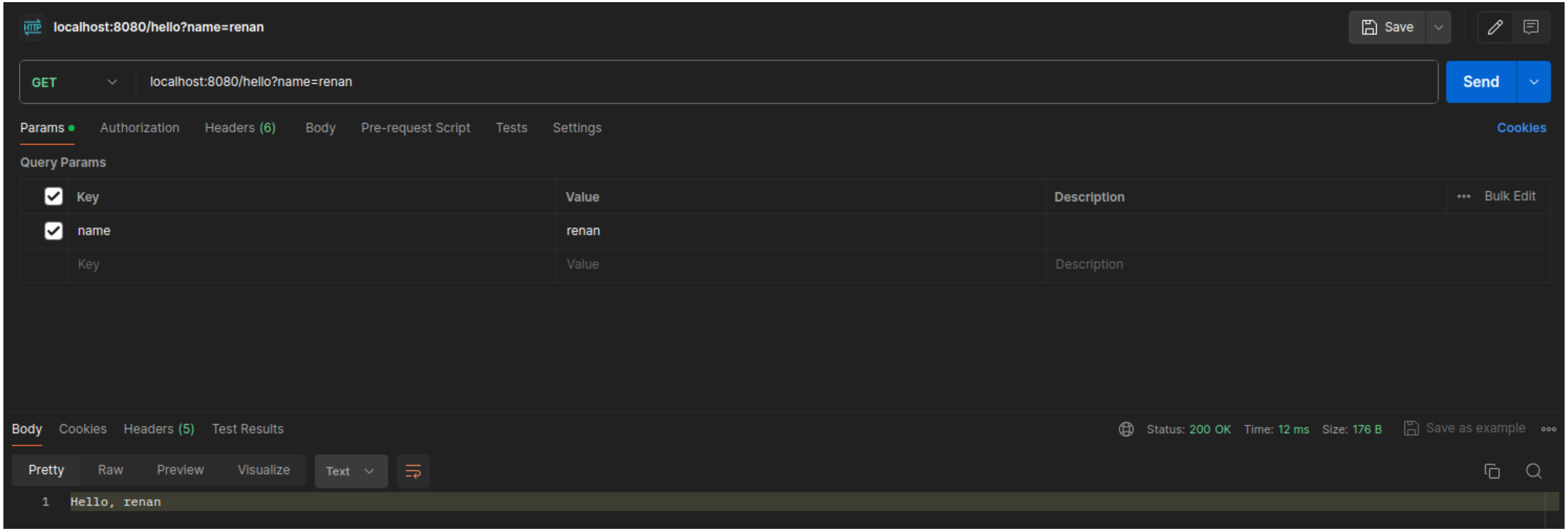
HTTP Server Requests Sum

GET [200] - /actuator/prometheus

HTTP Server Requests Max

GET [200] - /actuator/prometheus

- Se fizermos algumas requisições ao endpoint criado, vamos ver as métricas de acesso a ele no board



localhost:8080/hello?name=renan

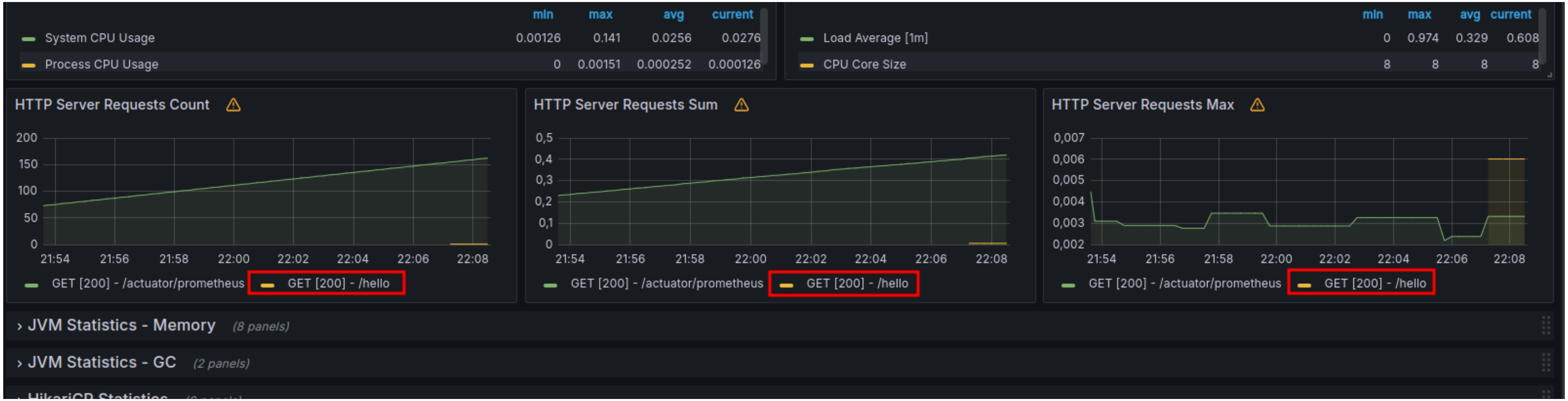
GET localhost:8080/hello?name=renan

Query Params

☑	Key	Value	Description
☑	name	renan	
	Key	Value	Description

Status: 200 OK Time: 12 ms Size: 176 B

```
Hello, renan
```



	min	max	avg	current
▬ System CPU Usage	0.00126	0.141	0.0256	0.0276
▬ Process CPU Usage	0	0.00151	0.000252	0.000126

	min	max	avg	current
▬ Load Average [1m]	0	0.974	0.329	0.608
▬ CPU Core Size	8	8	8	8

HTTP Server Requests Count ⚠

HTTP Server Requests Sum ⚠

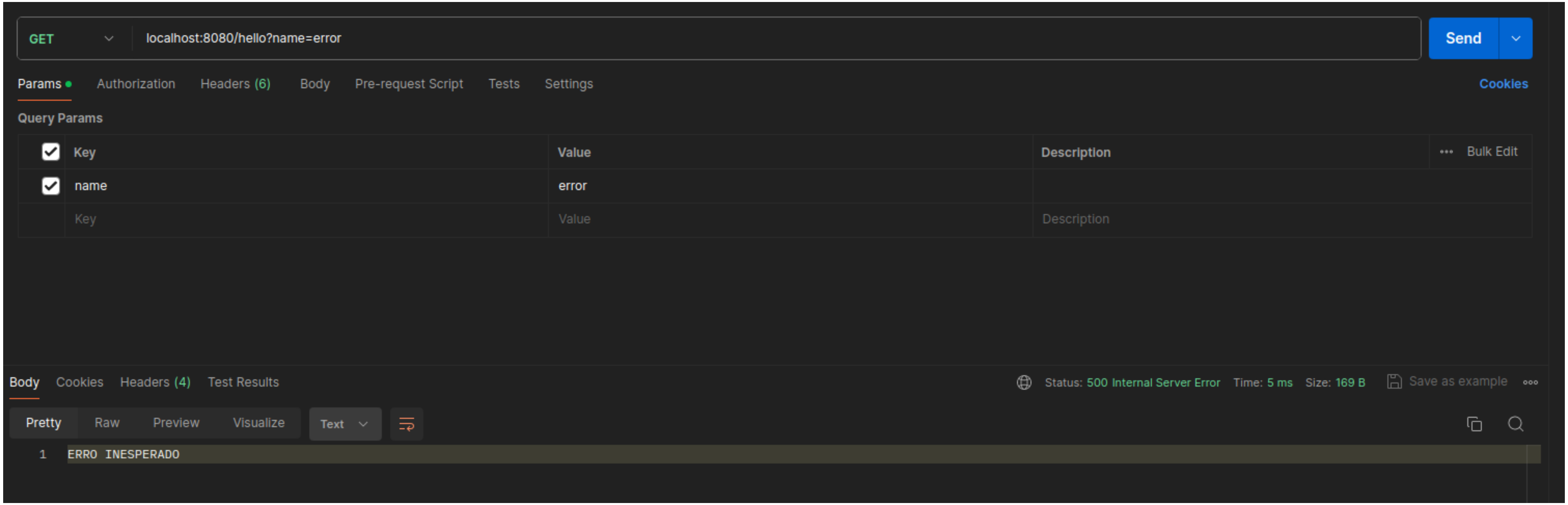
HTTP Server Requests Max ⚠

› JVM Statistics - Memory *(8 panels)*

› JVM Statistics - GC *(2 panels)*

› HikariCP Statistics

- Fazendo uma requisição com o parâmetro “error” para mudar o retorno http status.



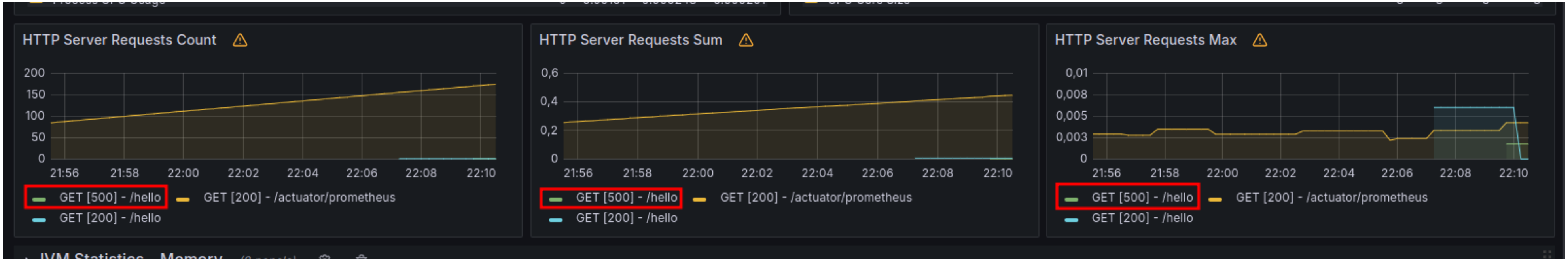
GET localhost:8080/hello?name=error

Query Params

	Key	Value	Description
☑	name	error	

Status: 500 Internal Server Error Time: 5 ms Size: 169 B

```
ERRO INESPERADO
```


HTTP Server Requests Count
200
150
100
50
0
21:56
21:58
22:00
22:02
22:04
22:06
22:08
22:10
GET [500] - /hello
GET [200] - /actuator/prometheus
GET [200] - /hello
HTTP Server Requests Sum
0,6
0,4
0,2
0
21:56
21:58
22:00
22:02
22:04
22:06
22:08
22:10
GET [500] - /hello
GET [200] - /actuator/prometheus
GET [200] - /hello
HTTP Server Requests Max
0,01
0,008
0,005
0,003
0
21:56
21:58
22:00
22:02
22:04
22:06
22:08
22:10
GET [500] - /hello
GET [200] - /actuator/prometheus
GET [200] - /hello
JVM Statistics - Memory