

Break New Ground

Break New Ground

Como monitorar métricas em ambientes distribuídos para garantir a disponibilidade dos seus serviços

Elder Moraes | @elderjava
Developer Advocate

Julho, 2019

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

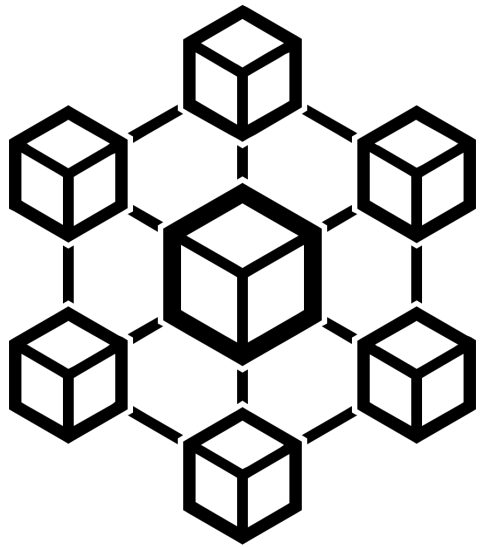
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.



“If I could change the world...”

– Eric Clapton



Métricas

Métrica é uma medida para avaliar,
controlar e/ou selecionar
quantitativamente: uma pessoa, um
processo, um evento ou uma
instituição

Métrica vs Health Check

RockyBalboaService



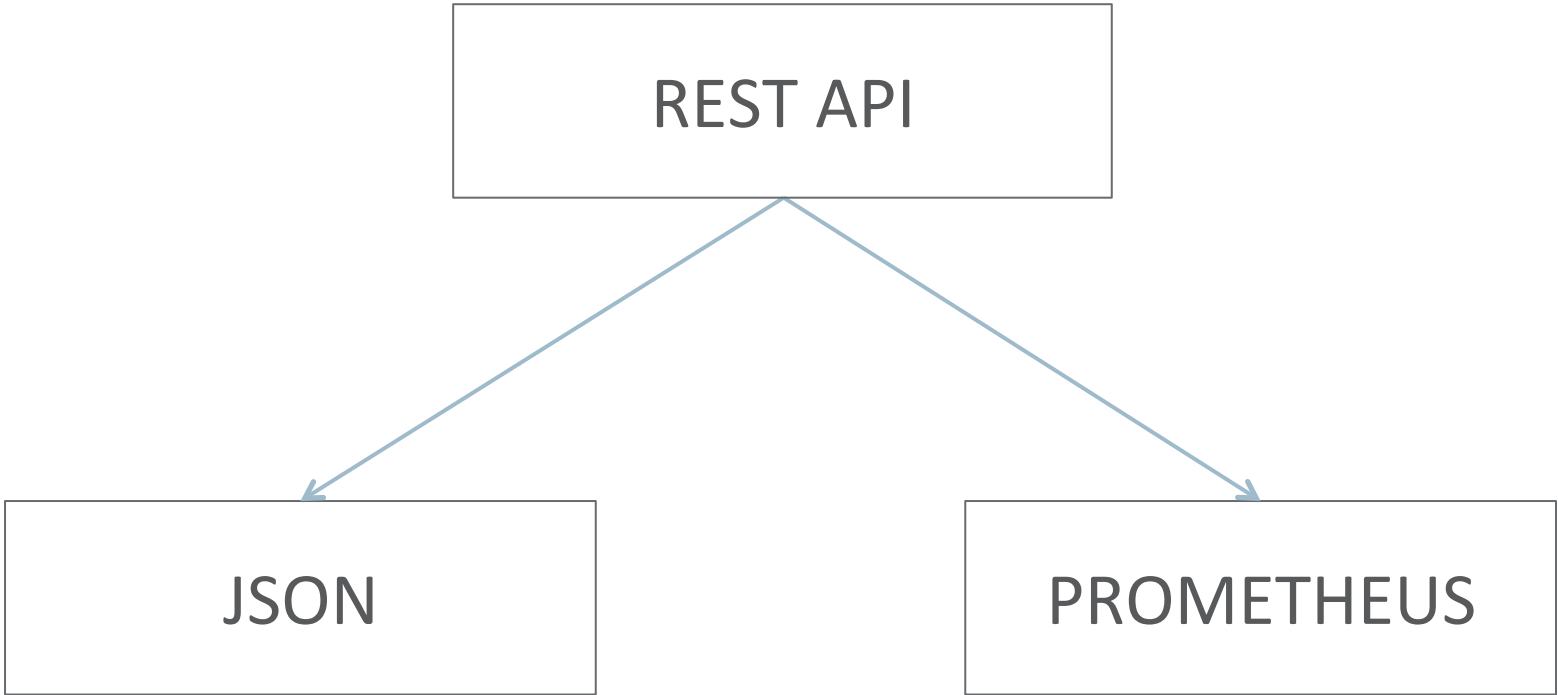
- Força do soco: 100%
- Sangramento: 0 ml/s
- Visão: 100%
- Nível suor: 1 ml/s
- Confiança: 10
- Pronúncia: Adrian!



- Força do soco: 0,5%
- Sangramento: 100 ml/s
- Visão: -10%
- Nível suor: 2000 ml/s (baldes)
- Confiança: 0
- Pronúncia: ãnhãããnnnn

Microprofile: 3 tipos de métricas

- base
 - JVM
 - Threads
 - Thread Pools
 - Classloading
 - SO
- vendor
- application



REST API

/metrics

/metrics/base

/metrics/vendor

/metrics/application

Formato Gauge JSON

```
{  
  "responsePercentage": 48.45632  
}
```

Formato Counter JSON

```
{  
  "hitCount": 45  
}
```

Formato Meter JSON

```
{
  "requests": {
    "count": 29382,
    "meanRate": 12.223,
    "oneMinRate": 12.563,
    "fiveMinRate": 12.364,
    "fifteenMinRate": 12.126,
  }
}
```


Formato Histogram JSON

```
{  
  "daily_value_changes": {  
    "count":2,  
    "min":-1624,  
    "max":26,  
    "mean":-799.0,  
    "stddev":825.0,  
    "p50":26.0,  
    "p75":26.0,  
    "p95":26.0,  
    "p98":26.0,  
    "p99":26.0,  
    "p999":26.0  
  }  
}
```

Formato Timer JSON

```
{
  "responseTime": {
    "count": 29382,
    "meanRate": 12.185627192860734,
    "oneMinRate": 12.563,
    "fiveMinRate": 12.364,
    "fifteenMinRate": 12.126,
    "min": 169916,
    "max": 5608694,
    "mean": 415041.00024926325,
    "stddev": 652907.9633011606,
    "p50": 293324.0,
    "p75": 344914.0,
    "p95": 543647.0,
    "p98": 2706543.0,
    "p99": 5608694.0,
    "p999": 5608694.0
  }
}
```

Todos possuem equivalentes no
formato Prometheus

Código orientado a métricas

Annotation

```
@Gauge(name = "queueSize")  
public int getQueueSize() {  
    return queue.size;  
}
```

Annotation	Aplica-se a	Descrição
@Counted	M, C, T	Contabiliza a quantidade de invocações do objeto anotado
@Gauge	M	Medidor que exibe o valor do objeto anotado
@Metered	M, C, T	Mede a frequência de invocações do objeto anotado
@Timed	M, C, T	Mede o tempo de duração do objeto anotado

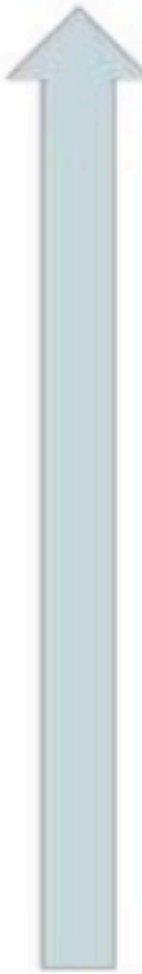
(C=Constructor, M=Method, T=Type)

Helidon



- É um conjunto de APIs para desenvolvimento de microservices
- Open Source (Apache 2.0)
- Helidon SE: microframework
- Helidon MP: MicroProfile

Larger



Smaller

Full-Stack



Spring Boot



Dropwizard

MicroProfile Based



Open Liberty



payara

THORNTAIL



helidon MP

Microframeworks



javalin



Spark



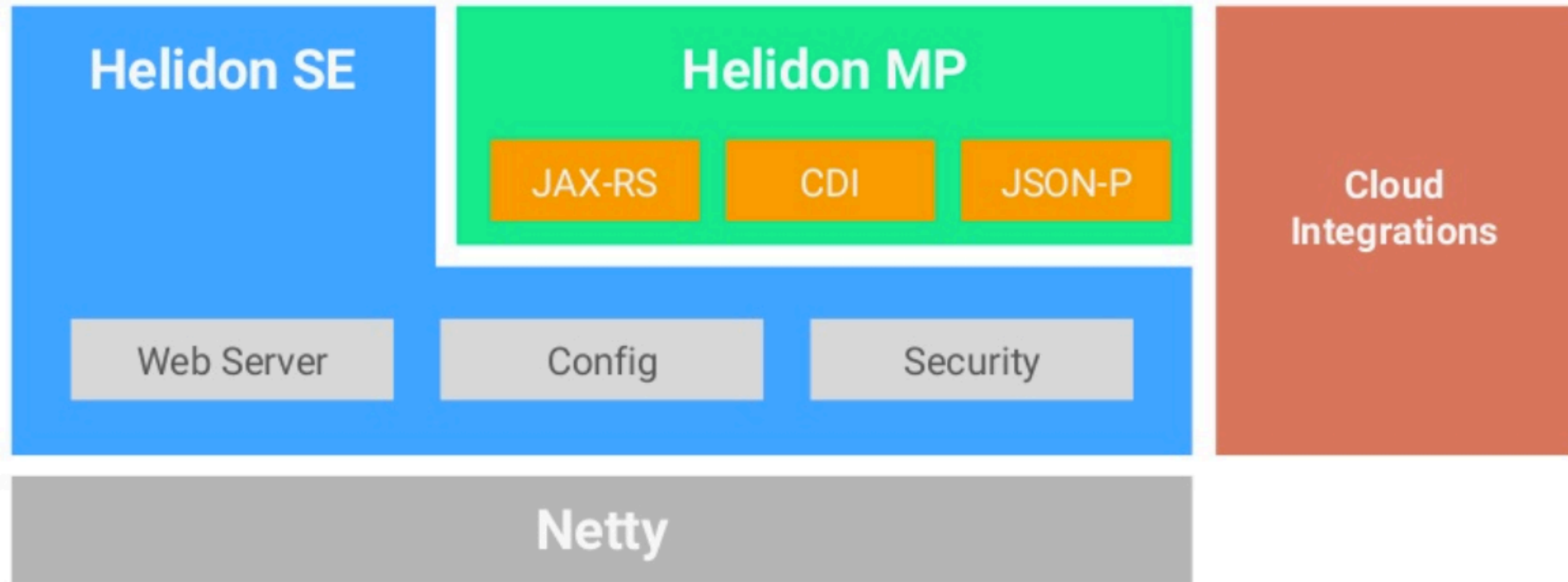
MICRONAUT™



helidon SE



Arquitetura



Código Helidon SE

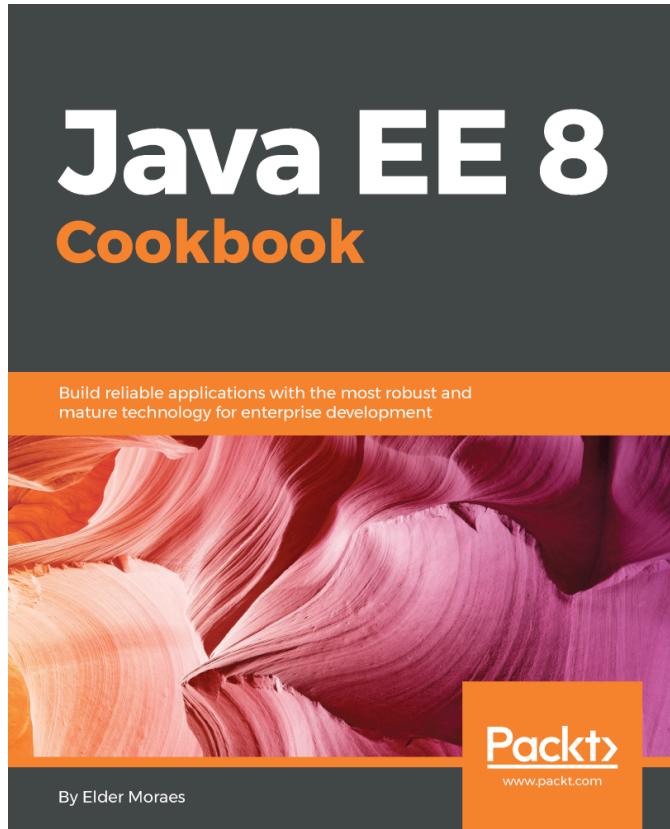
```
Routing routing = Routing.builder()  
    .get("/hello", (req, res) -> res.send("Hello World"))  
    .build();  
  
WebServer.create(routing)  
    .start();
```

Código Helidon MP

```
@Path("hello")
public class HelloWorld {
    @GET
    public String hello() {
        return "Hello World";
    }
}

Server.builder()
    .addResourceClass(HelloWorld.class)
    .build()
    .start();
```

DEMO TIME!!!



@elderjava

book.eldermoraes.com

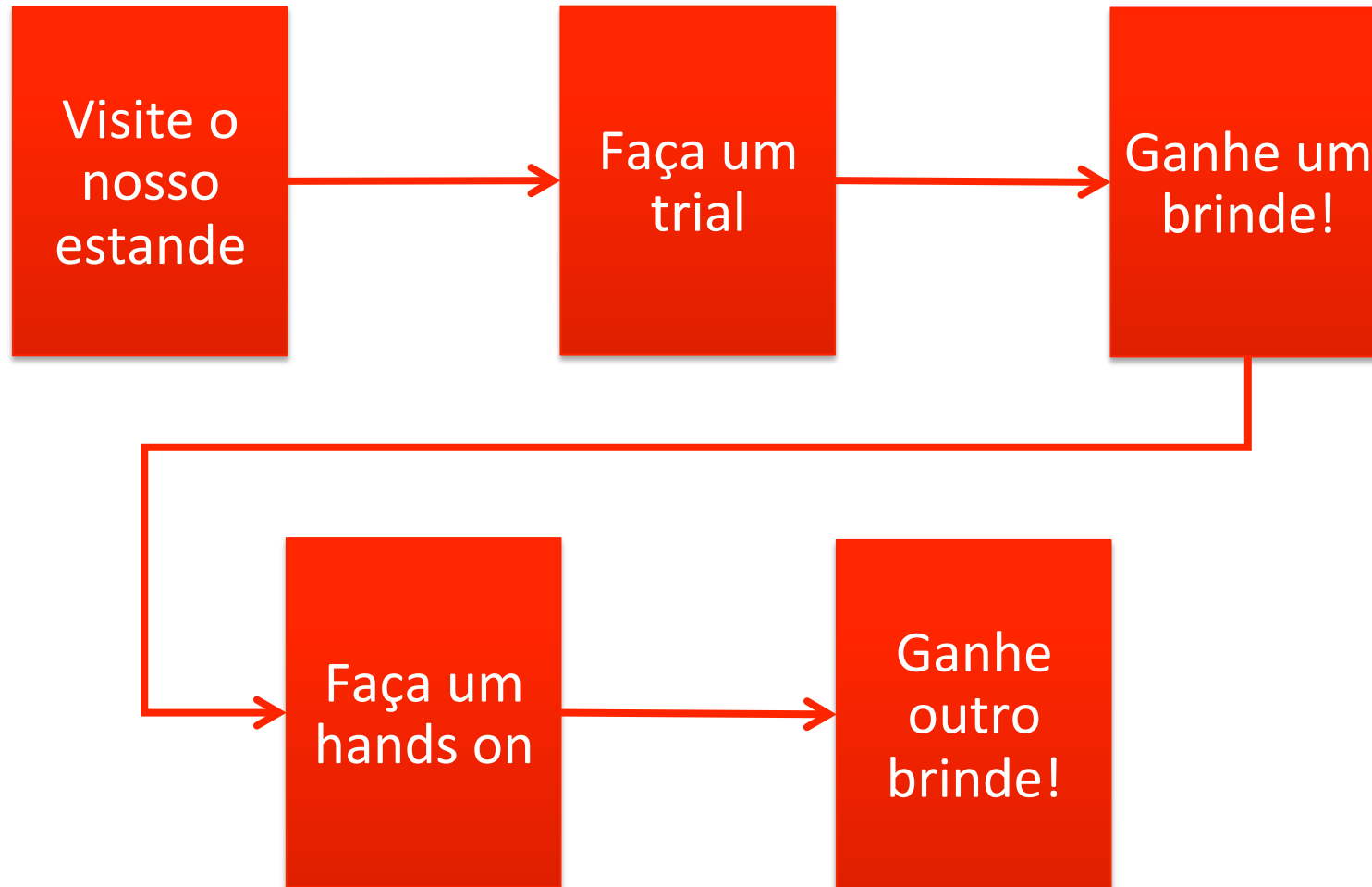
ORACLE®



Java @ Cloud Age

bit.ly/javacloudage





Break New Ground