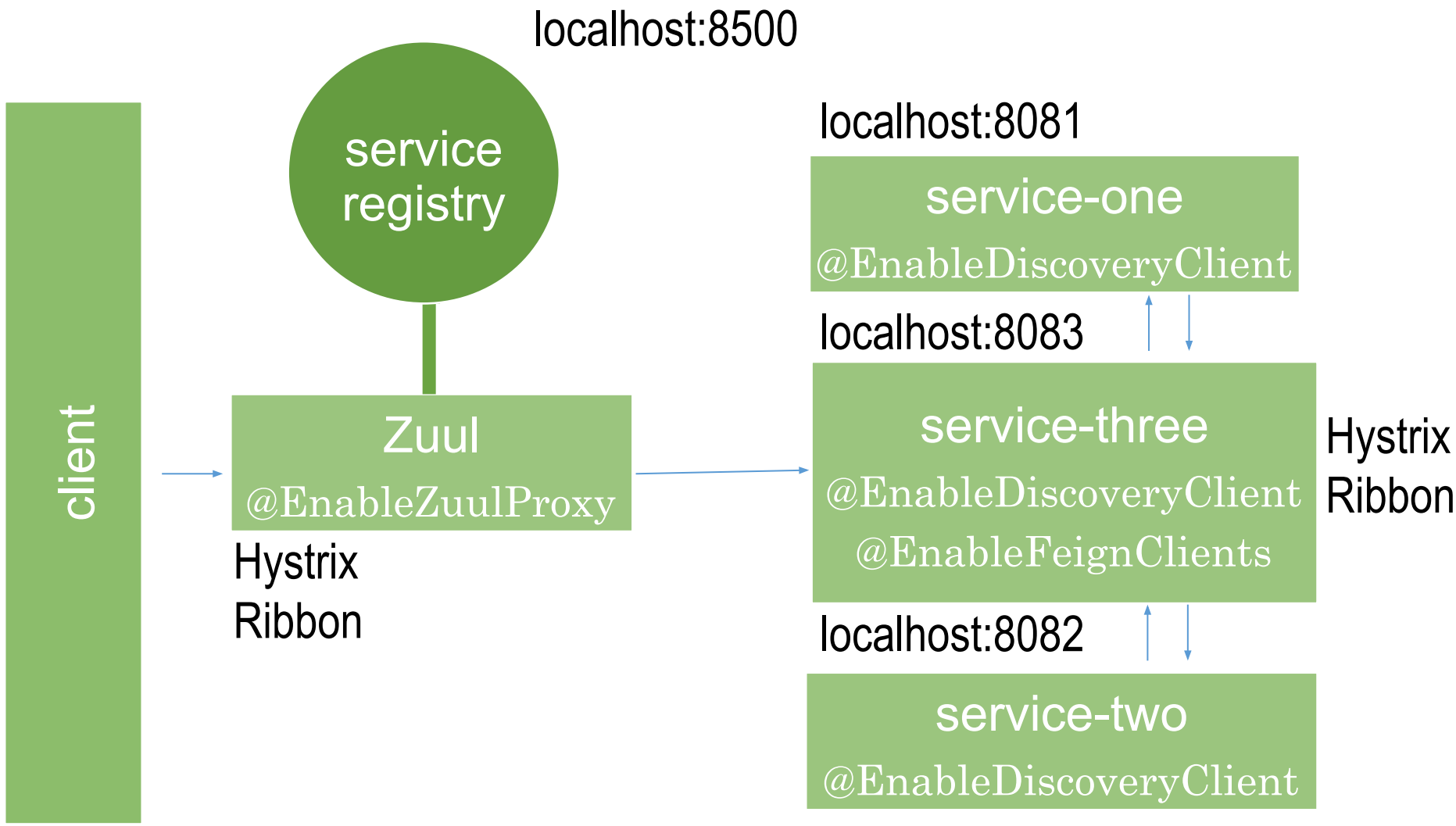
Microservices. Stage two. Tracing, Metrics.

by Dmitry Maronov,
Igor Zboychik

General

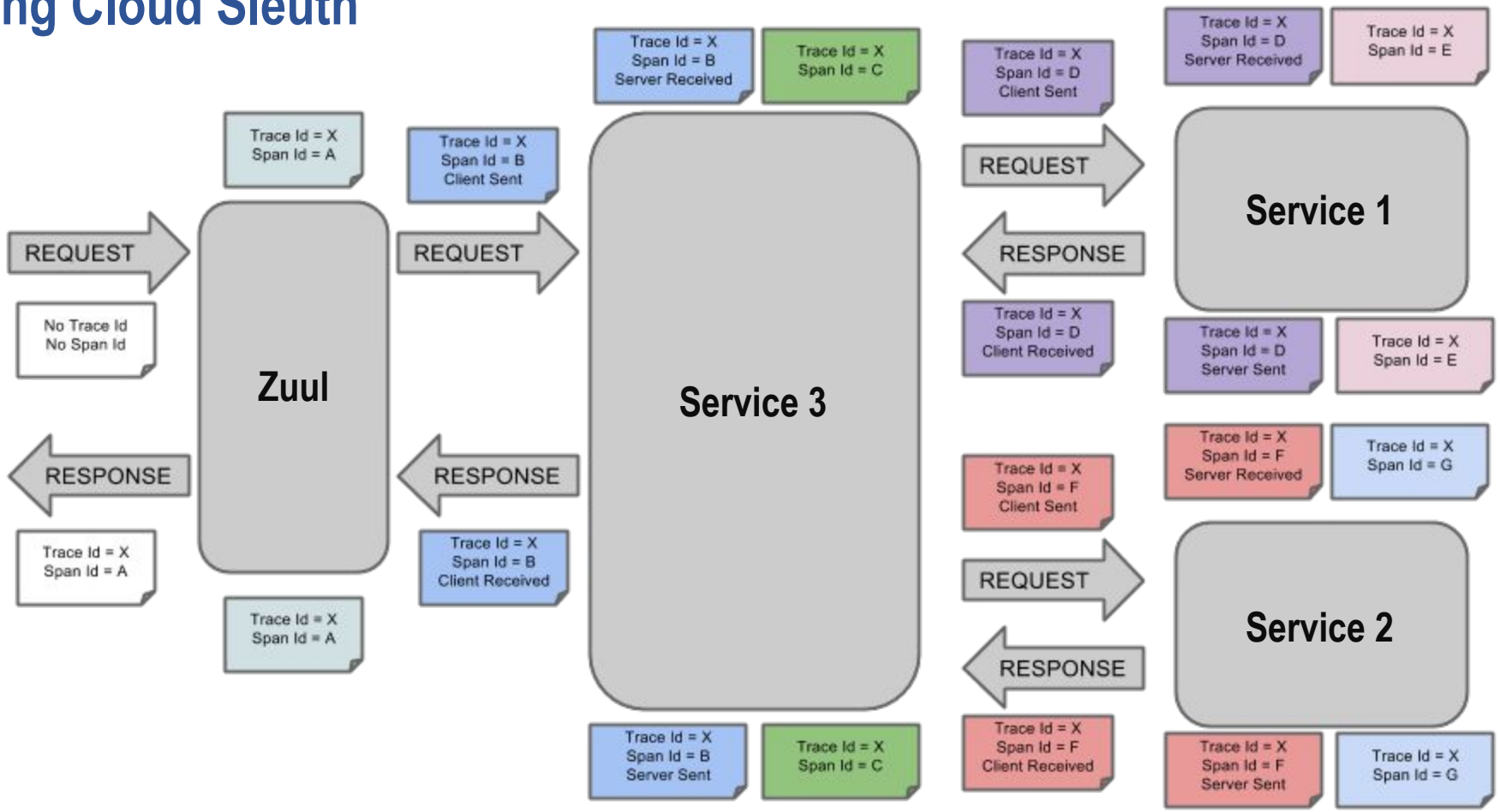
1. Sleuth;
2. Zipkin;
3. Spring Boot Actuator 2
4. Micrometer
5. Prometheus and Grafana

System overview



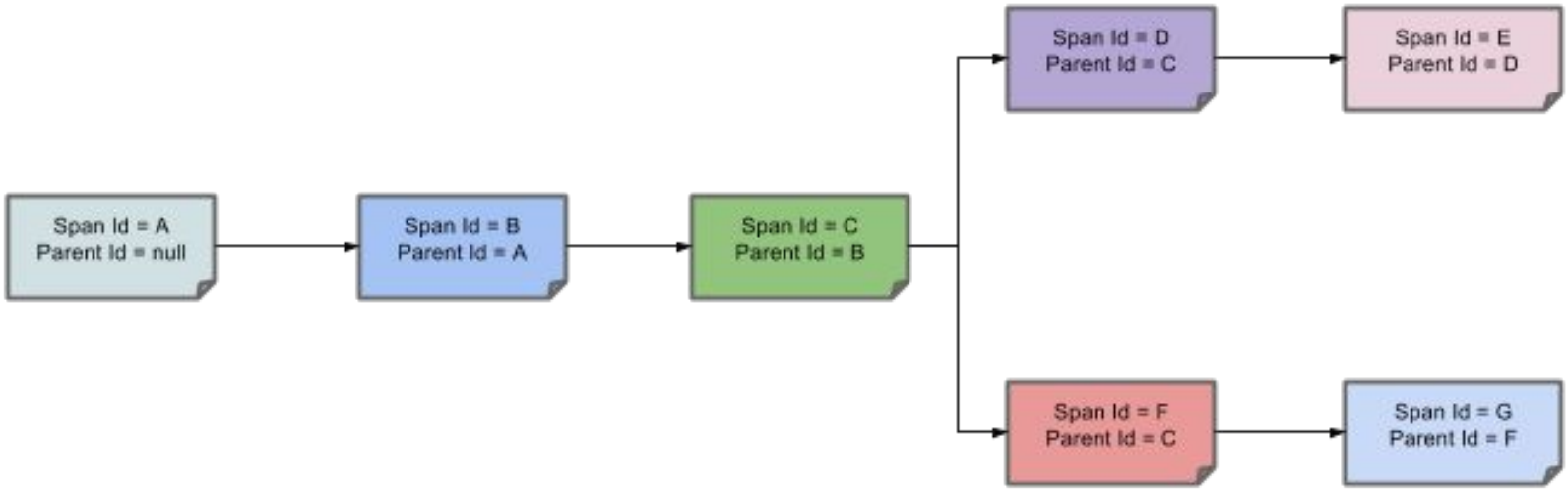
Tracing

Spring Cloud Sleuth



Tracing

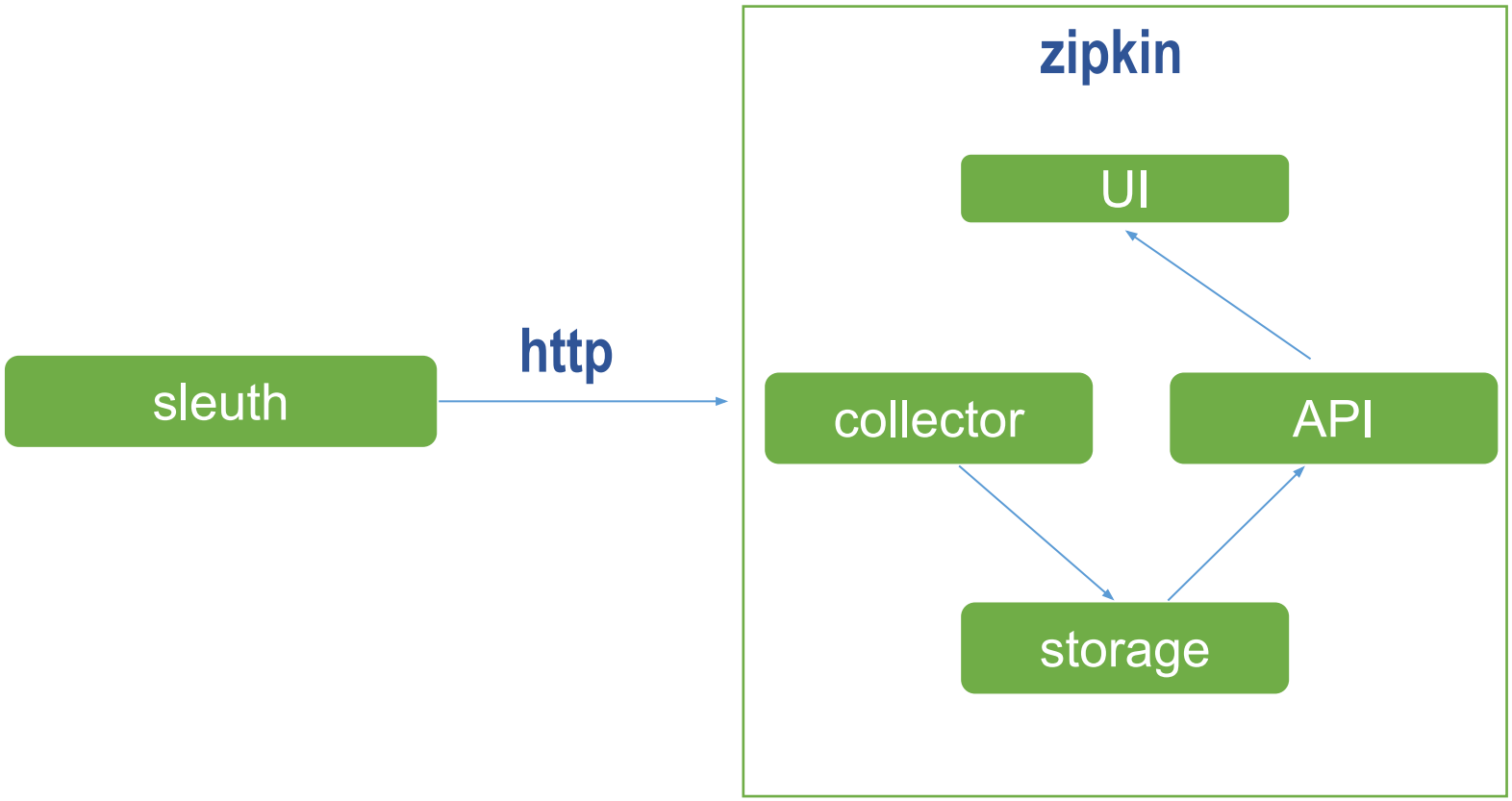
Spring Cloud Sleuth



- traceId;
- spanId;

Distributed Tracing with Zipkin

Zipkin



Infrastructure

Prometheus instance

Gateway

2 nodes of 1 service

Service registry

API:

/api/r/task?t=timeout http://51.15.124.217:8080/api/r/task?t=5

Spring Boot Actuator

Additional features to help you monitor and manage your application in production.

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

Spring Boot Actuator Technology Support

Spring Boot Actuator 1.x

- Spring MVC

Spring Boot Actuator 2.x

- Spring MVC
- Spring WebFlux
- Jersey

Spring Boot Actuator Endpoints

Supported Endpoints

/actuator/*

- beans
- conditions
- env
- health
- httptrace
- info
- metrics
- scheduledtasks
- sessions
- threaddump
- heapdump
- prometheus
- . . .

- renamed
- added

https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html

Spring Boot Actuator

How to enable endpoints

Default exposed endpoints are **/info** and **/health**

```
management:
  endpoints:
    web:
      exposure:
        include: *
```

```
management:
  endpoints:
    web:
      exposure:
        include: ["info","health","metrics","prometheus"]
```

All endpoints are **not secured** by default.

Spring Boot Actuator

Health Check

```yaml
management:
  endpoint:
    health:
      show-details: when_authorized
```

http://51.15.124.217:8080/actuator/health - admin/pass

Micrometer

Like SLF4J, but for Metrics

Instrument without vendor lock-in:

- Prometheus
- Netflix Atlas
- Datadog
- InfluxDB
- . . .

https://micrometer.io

Supported monitoring systems

Server polls	Client pushes	
Prometheus	Atlas, Datadog, Datadog StatsD, Influx, SignalFx, Telegraf StatsD, Wavefront, New Relic	Dimensional
	Graphite, Ganglia, JMX, Etsy StatsD, PCF Metrics 1.4	Hierarchical

jvm_memory_used_bytes{area="heap",id="PS Eden Space",} 8591592.0

metrics_name{tag="tag_value"} metrics_value

Prometheus

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

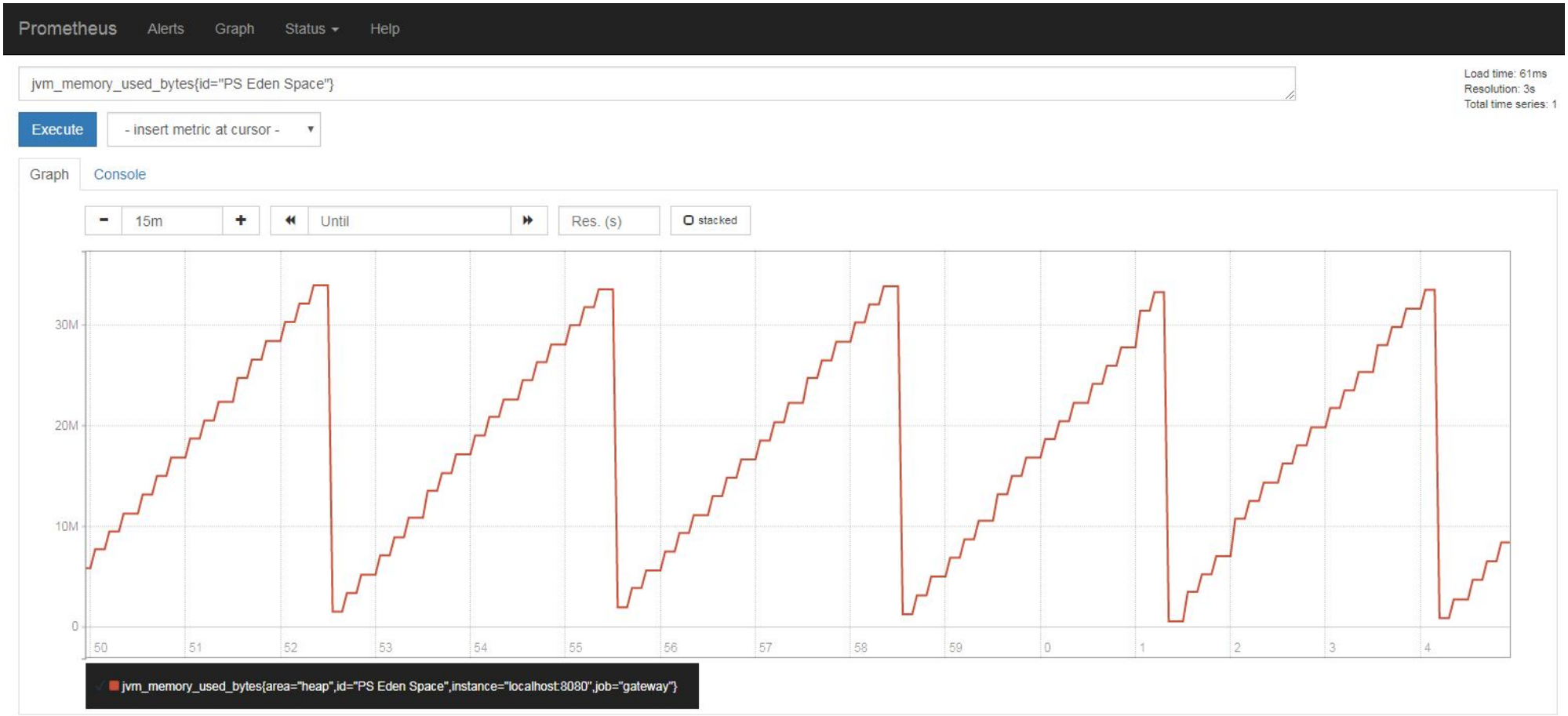
https://prometheus.io/

```
  GNU nano 2.5.3                File: /etc/prometheus/prometheus.yml

crape_configs:
  - job_name: 'gateway'
    metrics_path: '/actuator/prometheus'
    scrape_interval: 10s
    static_configs:
      - targets: ['localhost:8080']

global:
_ scrape_interval: 15s
```

http://51.15.124.217:8080/actuator/prometheus

Prometheus

http://51.15.124.217:9090/graph?g0.range_input=15m&g0.expr=jvm_memory_used_bytes%7Bid%3D%22PS%20Eden%20Space%22%7D&g0.tab=0

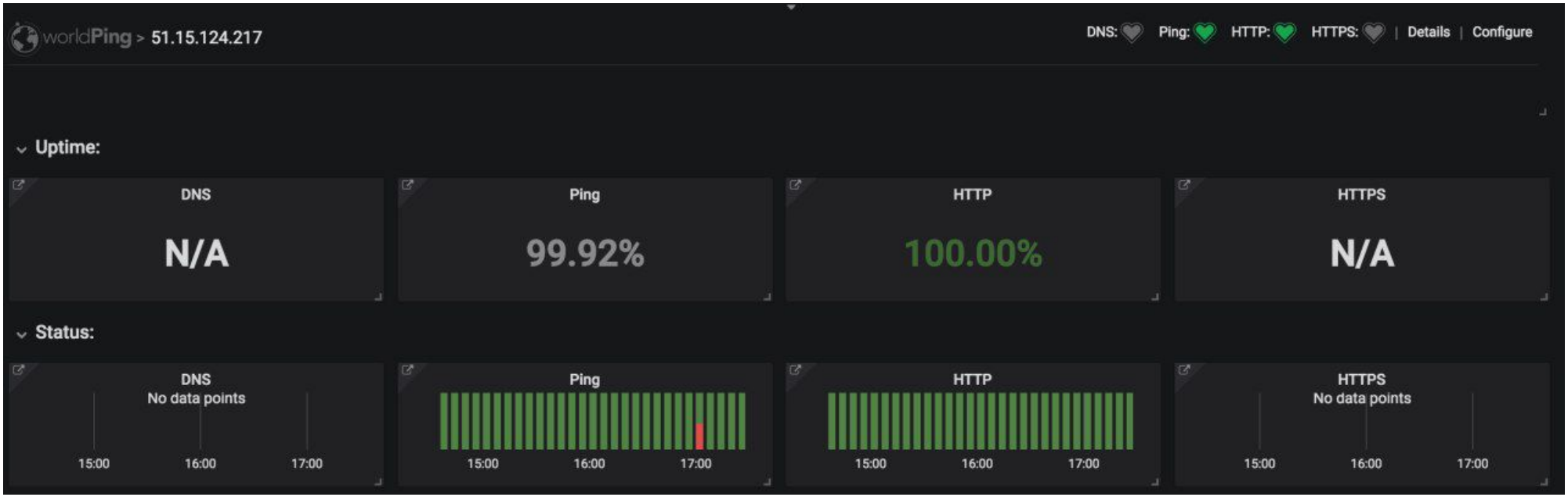


Grafana

https://grafana.com

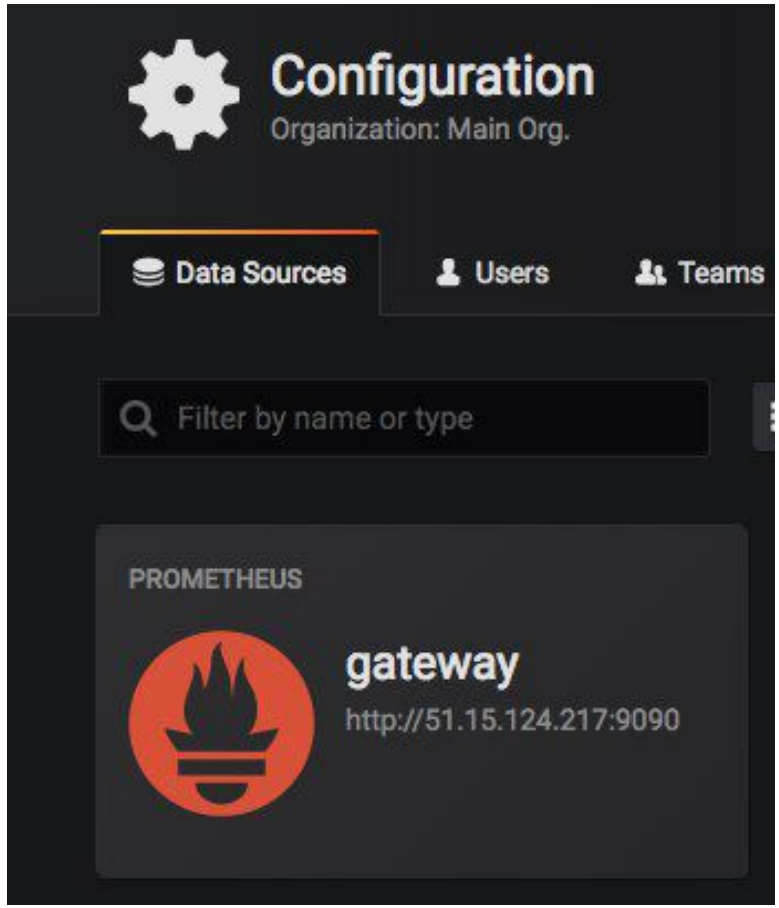
Free instance for 1 user with 5 dashboards

Plugins for http check with notifications



Grafana

Prometheus as datasource

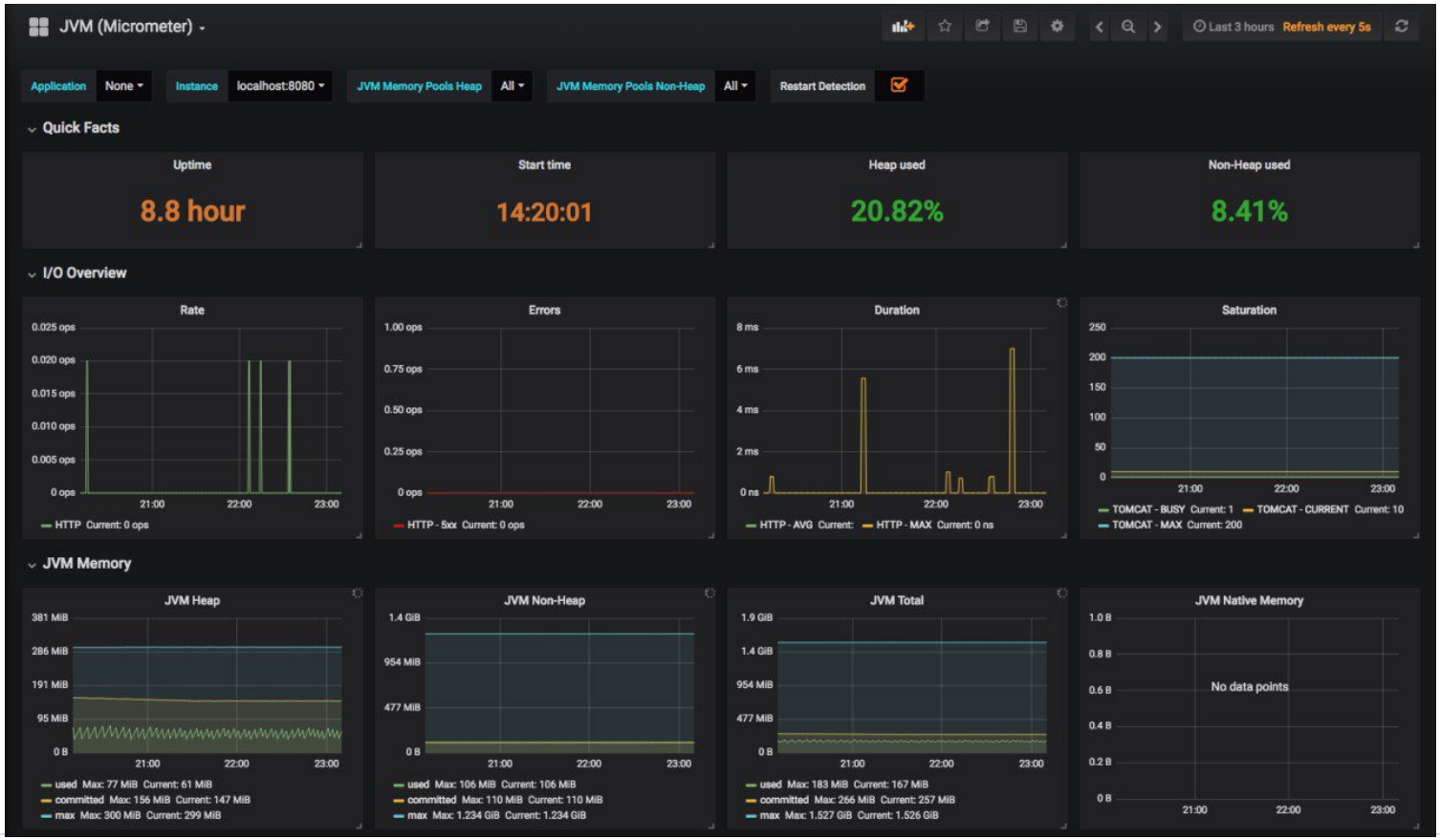


Grafana

Predefined dashboards

https://grafana.com/dashboards?dataSource=prometheus&search=java

https://grafana.com/dashboards/4701



Default HTTP metrics

Count, sum, max

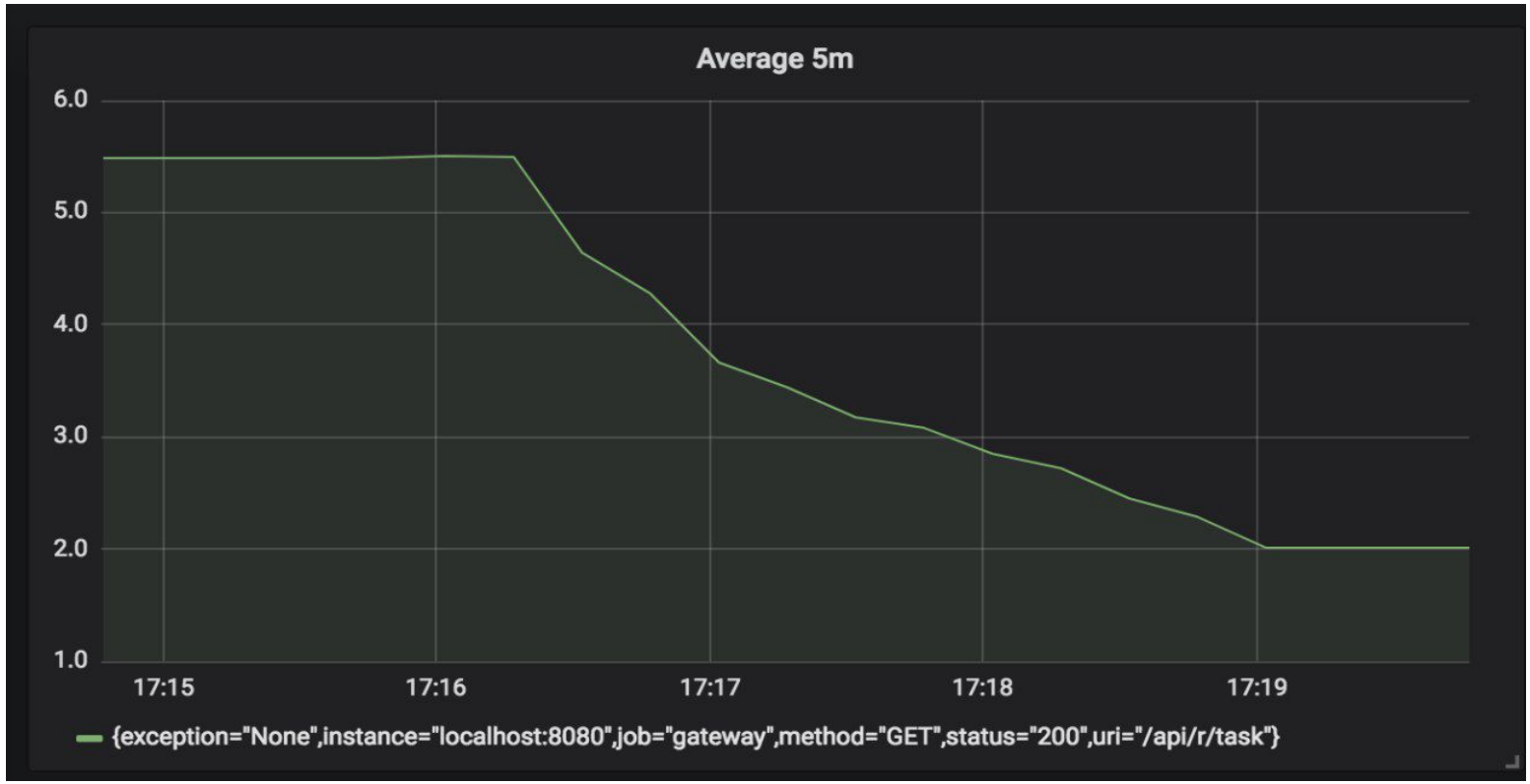
```
http_server_requests_seconds_count{exception="None",method="GET",status="500",uri="/api/r/task",} 41.0
http_server_requests_seconds_sum{exception="None",method="GET",status="500",uri="/api/r/task",} 19.503125608
http_server_requests_seconds_count{exception="None",method="GET",status="200",uri="/api/r/task",} 2.0
http_server_requests_seconds_sum{exception="None",method="GET",status="200",uri="/api/r/task",} 2.430160758
# HELP http_server_requests_seconds_max
# TYPE http_server_requests_seconds_max gauge
http_server_requests_seconds_max{exception="None",method="GET",status="500",uri="/api/r/task",} 0.032085282
http_server_requests_seconds_max{exception="None",method="GET",status="200",uri="/api/r/task",} 1.281689419
```

Default for zuul: uri="/api/r/**"

Custom WebMvcTagsProvider bean required

Average by 5 minutes

```
rate(http_server_requests_seconds_sum{status="200"}[5m]) /
rate(http_server_requests_seconds_count{status="200"}[5m])
```



https://prometheus.io/docs/practices/histograms/
https://prometheus.io/docs/prometheus/latest/querying/functions/#rate()

Histograms and percentiles

Histogram

```yaml
management:
  metrics:
    distribution:
      percentiles-histogram:
        http:
          server:
            requests: true
```

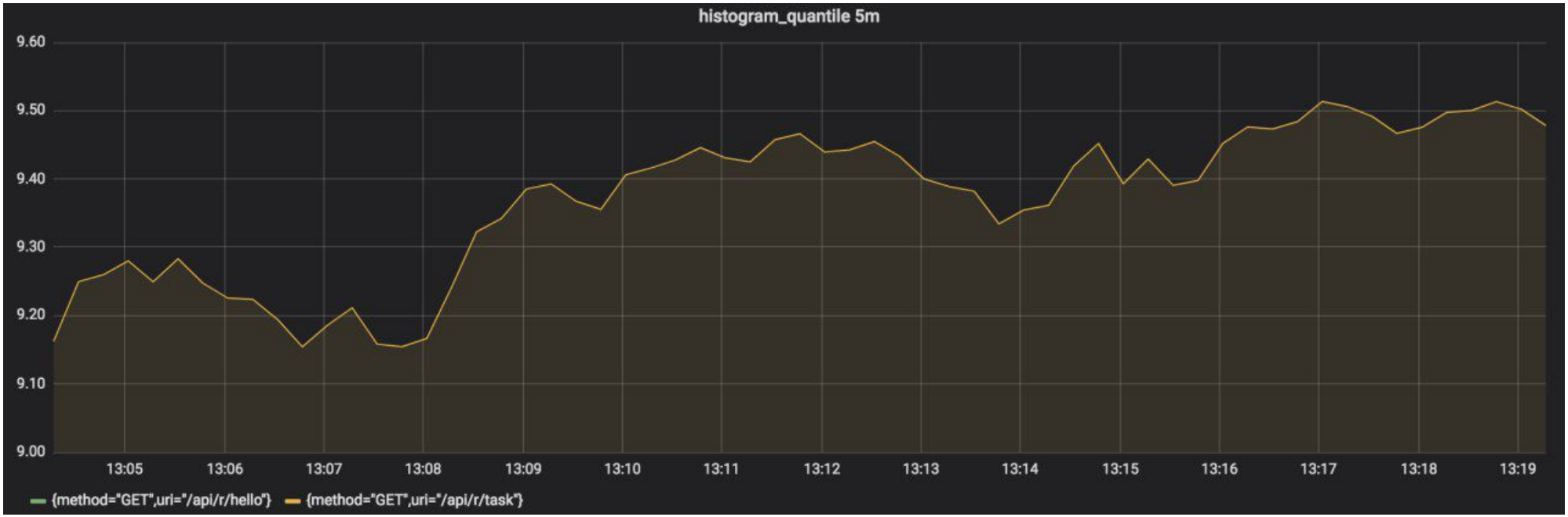
```
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="1.073741824",} 446.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="1.431655765",} 449.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="1.789569706",} 449.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="2.0",} 449.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="2.147483647",} 840.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="2.505397588",} 840.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="2.863311529",} 840.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="3.22122547",} 888.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="3.579139411",} 888.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="3.937053352",} 888.0
http_server_requests_seconds_bucket{exception="None",method="GET",status="200",uri="/api/r/task",le="4.294967296",} 957.0
```

The number of requests that took less than time in le

http://51.15.124.217:8080/actuator/prometheus

Query percentiles

histogram_quantile(0.9,
sum(rate(http_server_requests_seconds_bucket{status="200"}[5m]))
by (uri, method, le))



90 percentile of requests duration over the last 5m, aggregated by uri and method

SLA (Service Level Agreement)

```yaml
management:
  metrics:
    sla:
      http:
        server:
          requests: [100ms, 200ms, 400ms, 600ms, 800ms, 1s, 2s, 5s]
```

http://51.15.124.217:8080/actuator/prometheus

Monitor user experience by Apdex

Apdex = (Satisfied requests + Tolerating requests / 2)
/ Total numbers of requests

Satisfied: The response time is <= T.

Tolerating: > T && <= 4T

Frustrated: > 4T

https://docs.newrelic.com/docs/apm/new-relic-apm/apdex/apdex-measuring-user-satisfaction

Monitor user experience by Apdex

For example, T = 1.5 [sec]

Level	Multiplier	Time
Satisfied	T or less	<= 1.5 sec
Tolerating	> T, <= 4T	Between 1.5 and 6 sec
Frustrated	> 4T	> 6 sec

```
(
    sum(rate(http_server_requests_seconds_bucket{status="200", le="1.5"}[2m])) by (uri, method)
    +
    ((sum(rate(http_server_requests_seconds_bucket{status="200", le="6.0"}[2m])) by (uri, method) - sum(rate(http_server_requests_seconds_bucket{status="200", le="1.5"}[2m])) by (uri, method)) / 2)
)
/ sum(rate(http_server_requests_seconds_count{status="200"}[2m])) by (uri, method)
```

https://prometheus.io/docs/practices/histograms/#apdex-score

Apdex results

From	To	Result
0.94	1.00	Excellent
0.85	0.94	Good
0.70	0.85	Fair
0.50	0.70	Poor
0.00	0.50	Unacceptable

Client-side Percentile

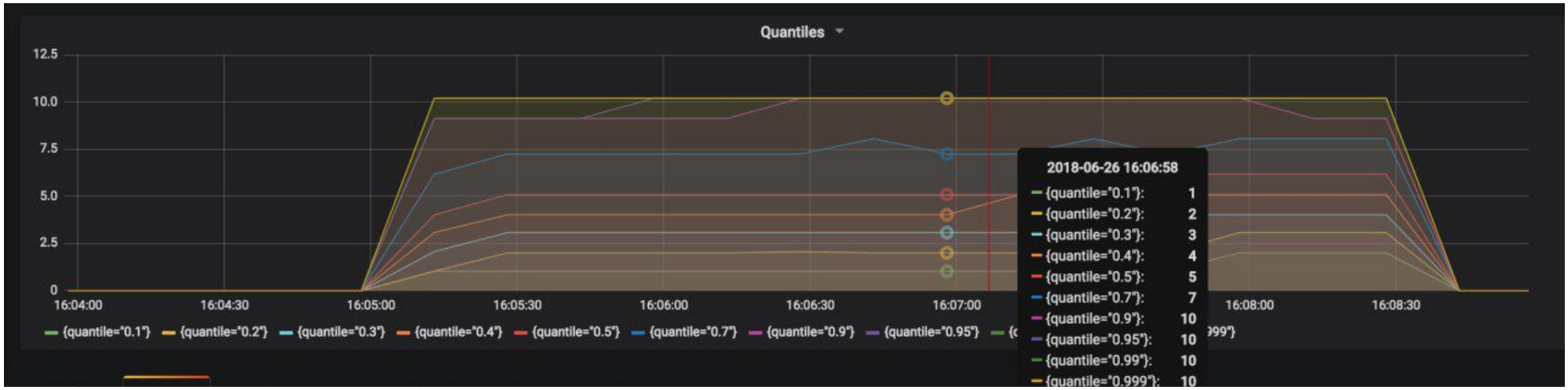
```yaml
management:
  metrics:
    percentiles:
      http:
        server:
          requests: [0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 0.95, 0.99, 0.999]
```

```
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.1",} 2.012217344
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.2",} 2.012217344
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.3",} 3.085959168
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.4",} 4.025483264
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.5",} 5.099225088
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.7",} 7.246708736
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.9",} 9.125756928
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.95",} 10.199498752
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.99",} 10.199498752
http_server_requests_seconds{exception="None",method="GET",status="200",uri="/api/r/task",quantile="0.999",} 10.199498752
```

0.5 quantile = median
50% of requests time are lower then 5.0992
and 50% are greater

Client-side Percentile in Grafana

max(http_server_requests_seconds{uri="/api/r/task", status="200", exception="None"}) by (quantile)



Micrometer API

More Counters and Timers

http://micrometer.io/docs/registry/prometheus

URL's

- https://zipkin.io/
- https://github.com/openzipkin/zipkin/tree/master/zipkin-server
- http://cloud.spring.io/spring-cloud-sleuth/single/spring-cloud-sleuth.html
- https://habr.com/company/jugru/blog/341026/
- https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html
- https://grafana.com/dashboards/4701
- https://prometheus.io/docs/practices/histograms/
- https://docs.newrelic.com/docs/apm/new-relic-apm/apdex/apdex-measuring-user-satisfaction
- http://micrometer.io/docs/registry/prometheus
- https://prometheus.io/docs/prometheus/latest/querying/functions
- http://51.15.124.217:9090/graph
- http://51.15.124.217:8080/actuator/health - admin/pass; /actuator/prometeus
- http://gl.igorz.me/coherent/boot-rest-grpc, https://github.com/zboigor/boot2metrics

Join us

Dmitry Maronov

Email DmitryMaronov@coherentsolutions.com

Skype csi.dmitrymaronov

Igor Zboychik

Email IgorZboychik@coherentsolutions.com

Skype csi.igorzboychik

www.coherentsolutions.com

COHERENT SOLUTIONS

Coherent Solutions

Better Teams. Better Results.