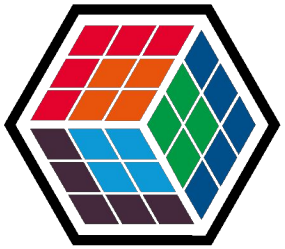


THE DEVELOPER'S CONFERENCE

Trilha – Python

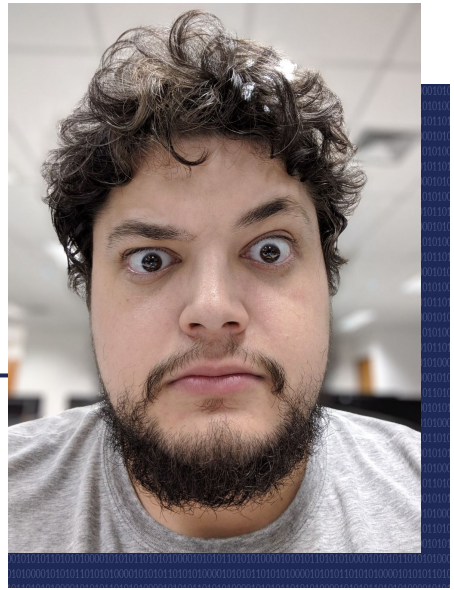
Bruno Gomes França e Yuri Piratello
Chapter Lead e Dev Senior na Stoodi

Quem somos?

Bruno França

Chapter Lead no Stoodi

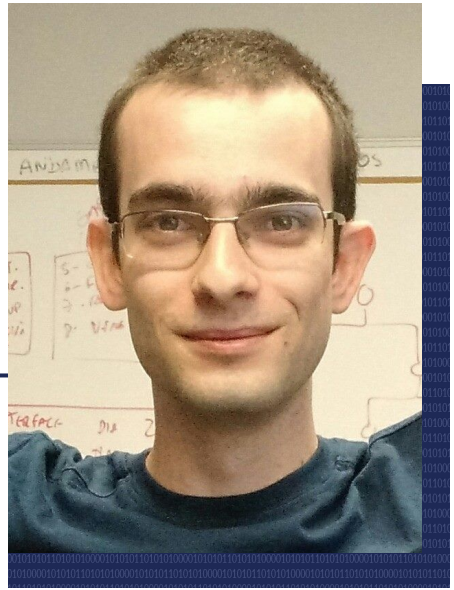
in/brunogomesfranca

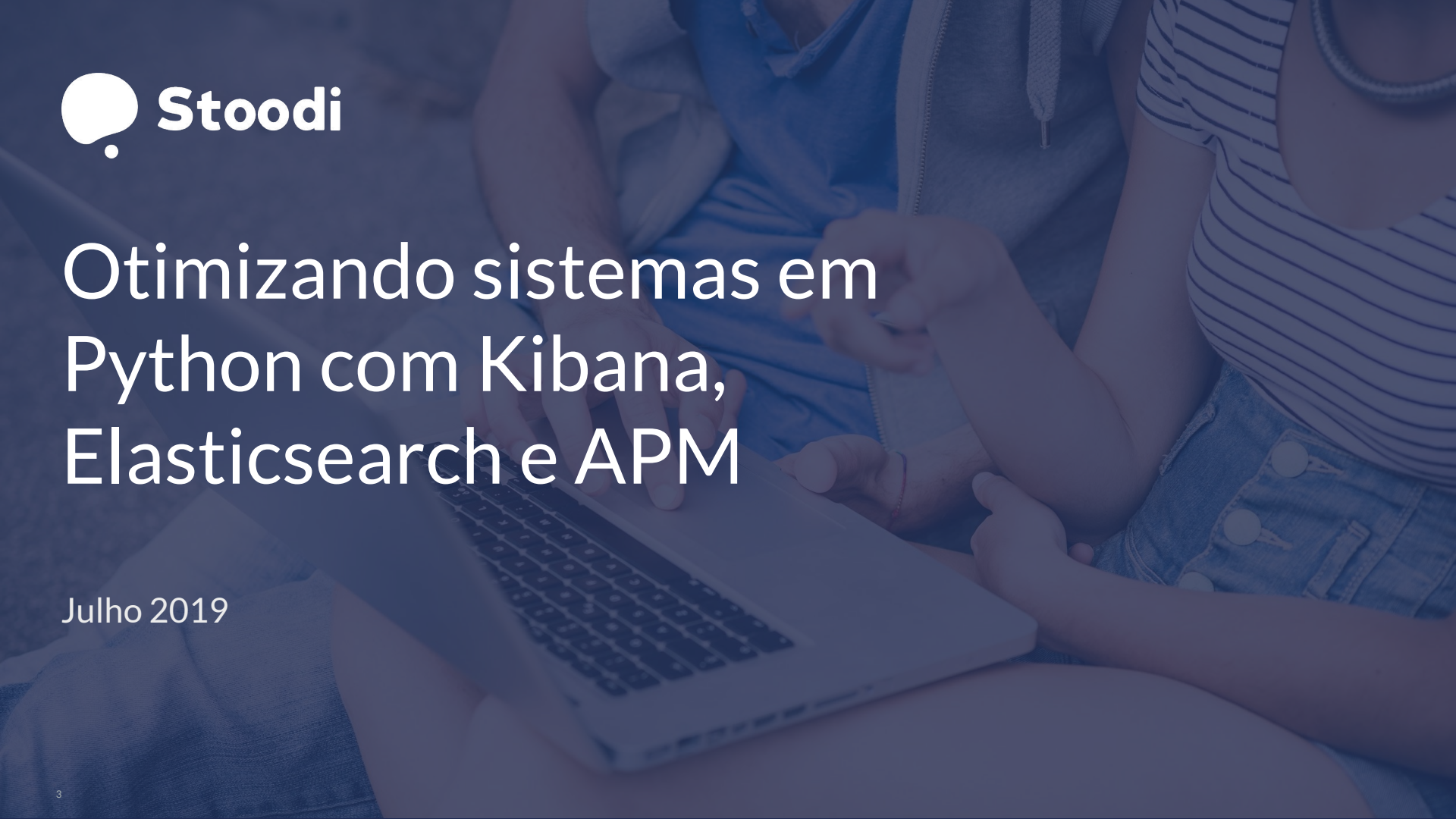


Yuri Piratello

Dev Sênior no Stoodi

in/yuripiratello



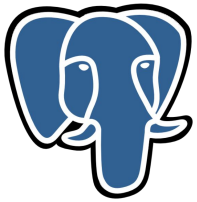


Stoodi

Otimizando sistemas em Python com Kibana, Elasticsearch e APM

Julho 2019

Nossa Stack



PostgreSQL

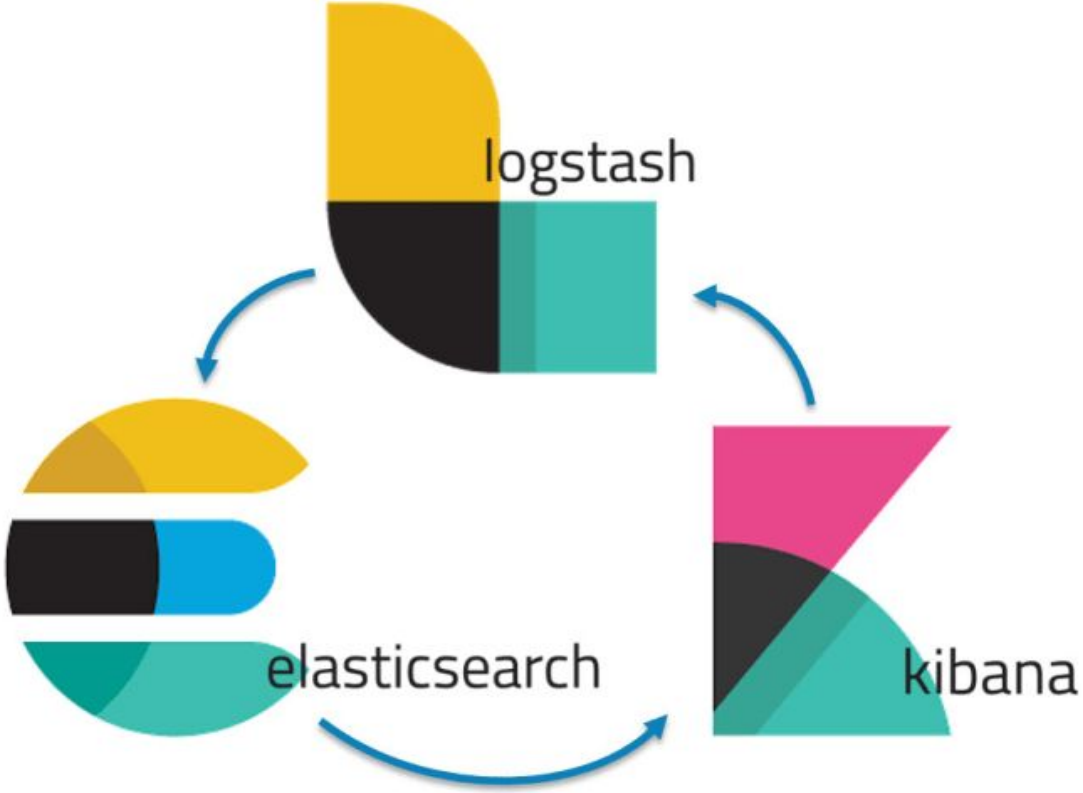
elasticsearch

redis

DynamoDB

E mais...

Monitoria, quase um...



Stack de Monitoria

Elasticsearch - Engine de pesquisa e análise.

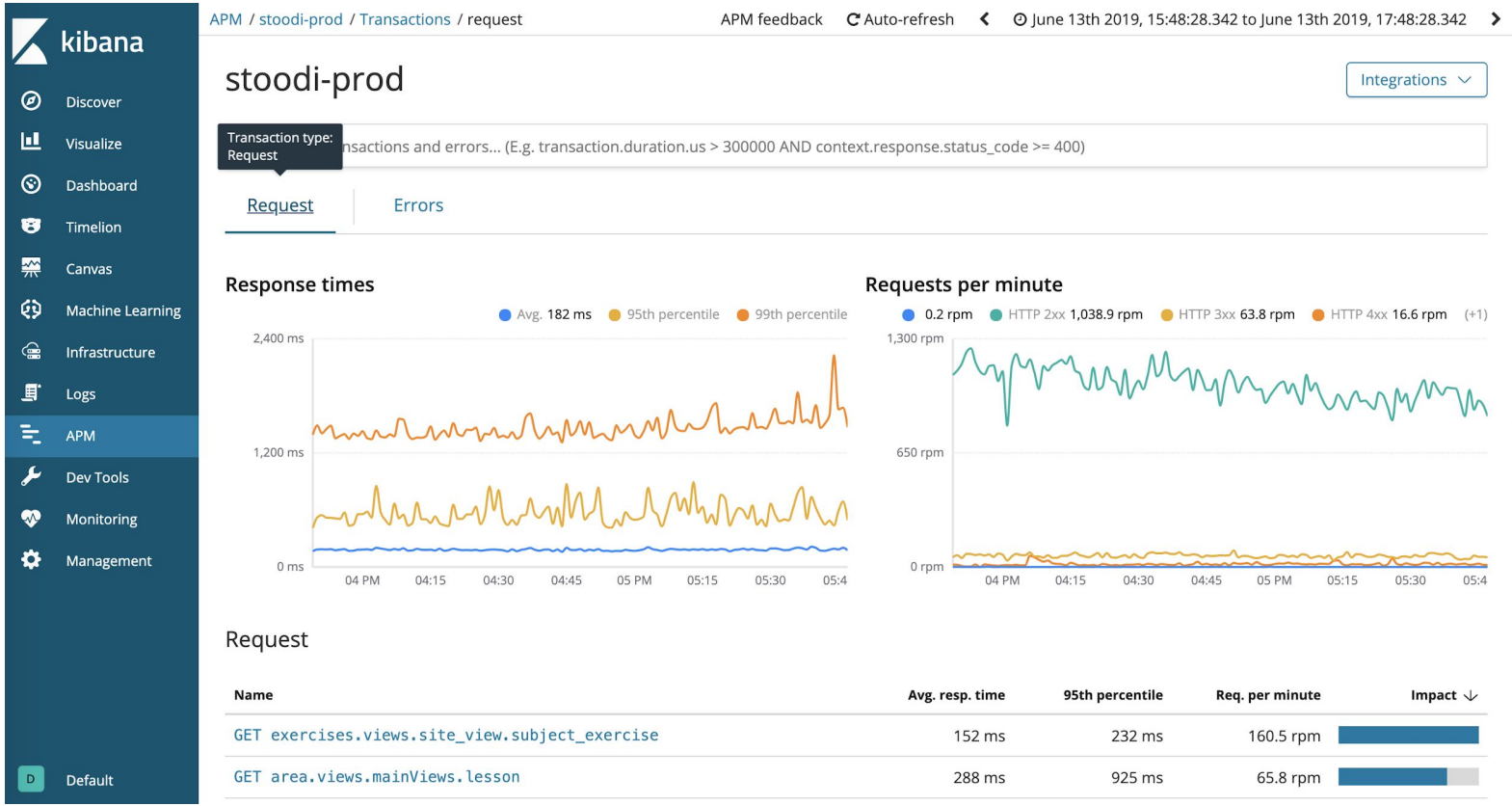
Kibana - Interface visual para acesso aos dados do Elasticsearch.

APM - Gestão de desempenho de aplicações (Application Performance Management).

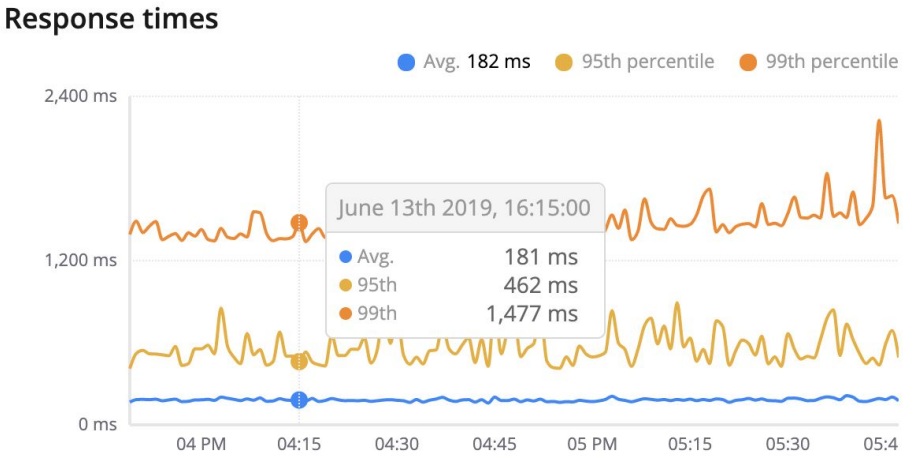
E agora?

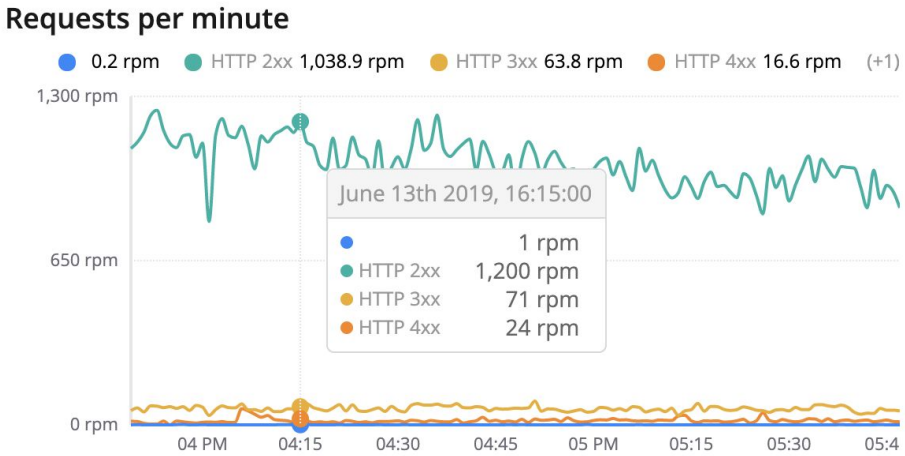
Ok, mas e como isso funciona?

Kibana


kibana
Discover
Visualize
Dashboard
Timelion
Canvas
Machine Learning
Infrastructure
Logs
APM
Dev Tools
Monitoring
Management
Default
APM / stoodi-prod / Transactions / request
APM feedback
Auto-refresh
June 13th 2019, 15:48:28.342 to June 13th 2019, 17:48:28.342
stoodi-prod
Integrations
Transaction type: Request
nsactions and errors... (E.g. transaction.duration.us > 300000 AND context.response.status_code >= 400)
Request
Errors
Response times
Avg. 182 ms
95th percentile
99th percentile
2,400 ms
1,200 ms
0 ms
04 PM
04:15
04:30
04:45
05 PM
05:15
05:30
05:4
Requests per minute
0.2 rpm
HTTP 2xx 1,038.9 rpm
HTTP 3xx 63.8 rpm
HTTP 4xx 16.6 rpm
(+1)
1,300 rpm
650 rpm
0 rpm
Request
Name
Avg. resp. time
95th percentile
Req. per minute
Impact
GET exercises.views.site_view.subject_exercise
152 ms
232 ms
160.5 rpm
GET area.views.mainViews.lesson
288 ms
925 ms
65.8 rpm

Kibana - Gráficos





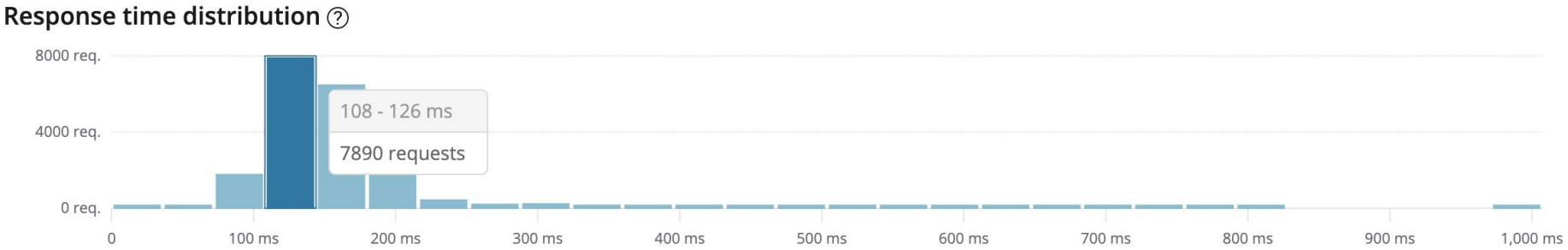
Kibana - Lista de Endpoints

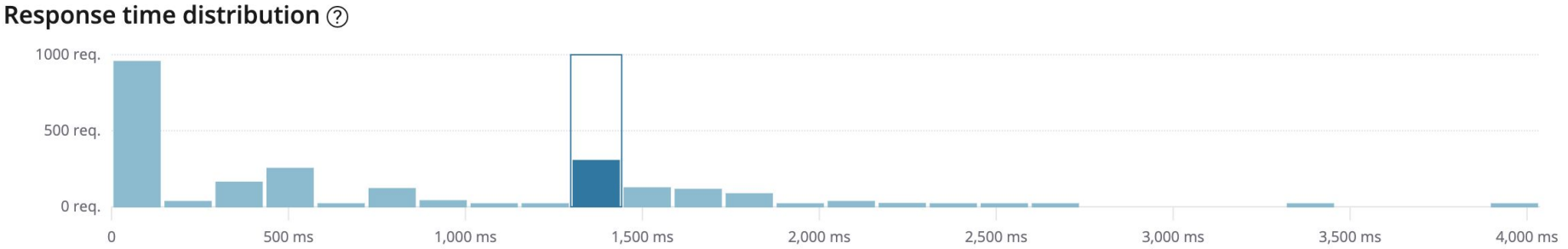
Request

Name	Avg. resp. time	95th percentile	Req. per minute	Impact ↓
GET exercises.views.site_view.subject_exercise	152 ms	232 ms	160.5 rpm	
GET area.views.mainViews.lesson	288 ms	925 ms	65.8 rpm	
GET exercises.views.site_view.video_modal	1,100 ms	2,301 ms	15.9 rpm	
POST area.views.mainViews.can_watch_this_video	266 ms	1,392 ms	59.3 rpm	
GET video.views.siteView.video	676 ms	1,799 ms	19.3 rpm	
GET main_page.views.generalViews.index	173 ms	294 ms	52.0 rpm	
GET exercises.views.site_view.subject_exercise_list	177 ms	272 ms	50.3 rpm	
GET subjectPanel.views.areas	233 ms	492 ms	31.7 rpm	
GET area.views.mainViews.lesson_exercise	187 ms	299 ms	37.5 rpm	
GET exercises.views.site_view.exercise	116 ms	171 ms	59.5 rpm	
GET subjectPanel.views.area	252 ms	399 ms	24.9 rpm	

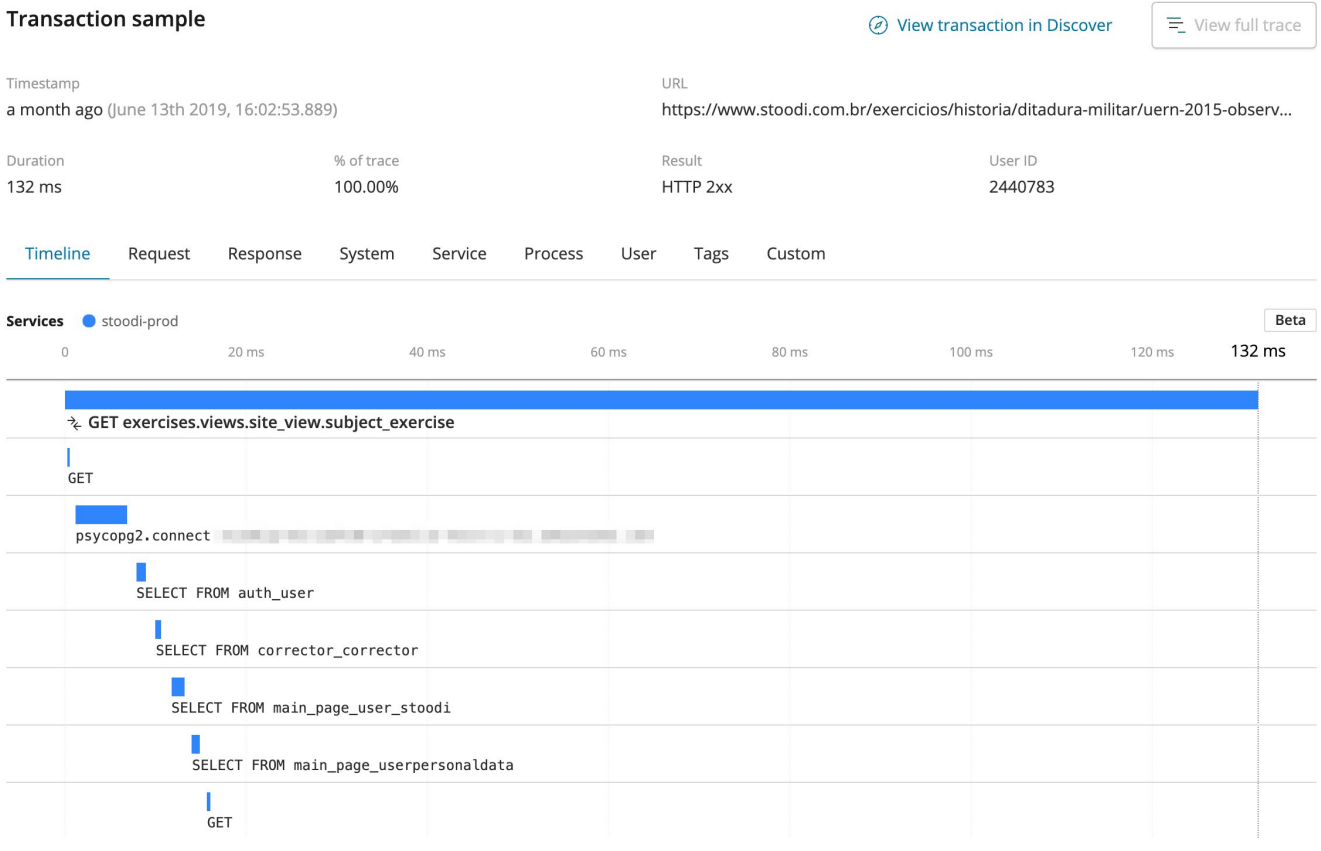
Kibana - Detalhes Endpoint

Kibana - Response time distribution

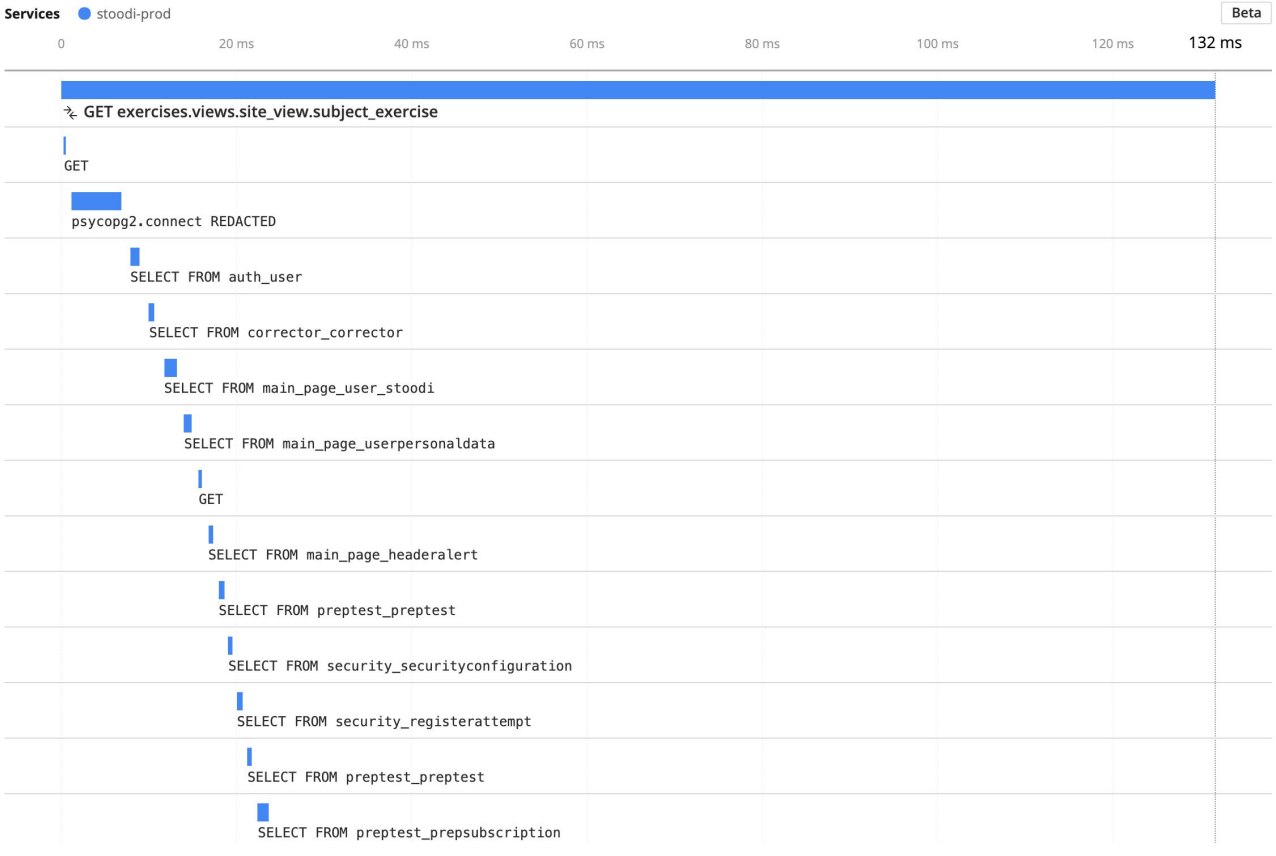




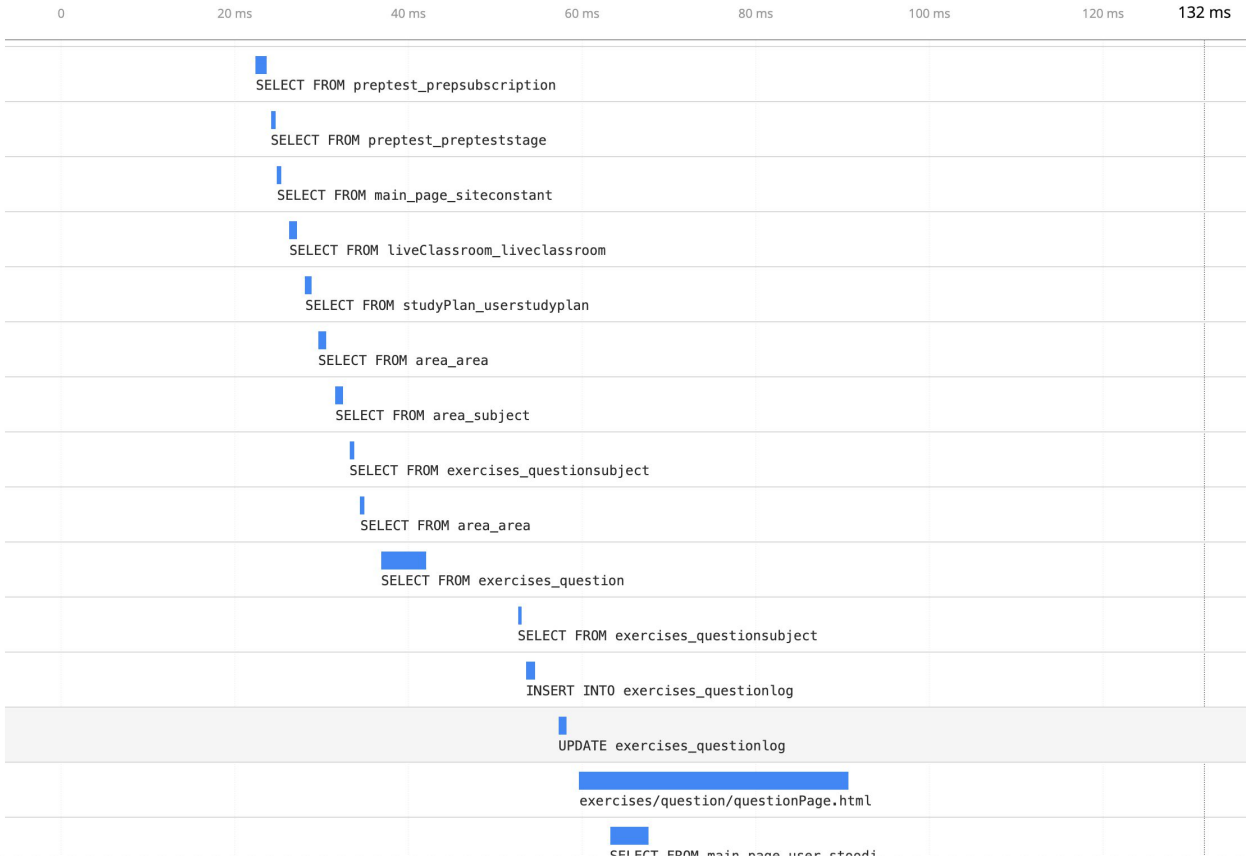
Kibana - Transaction Sample



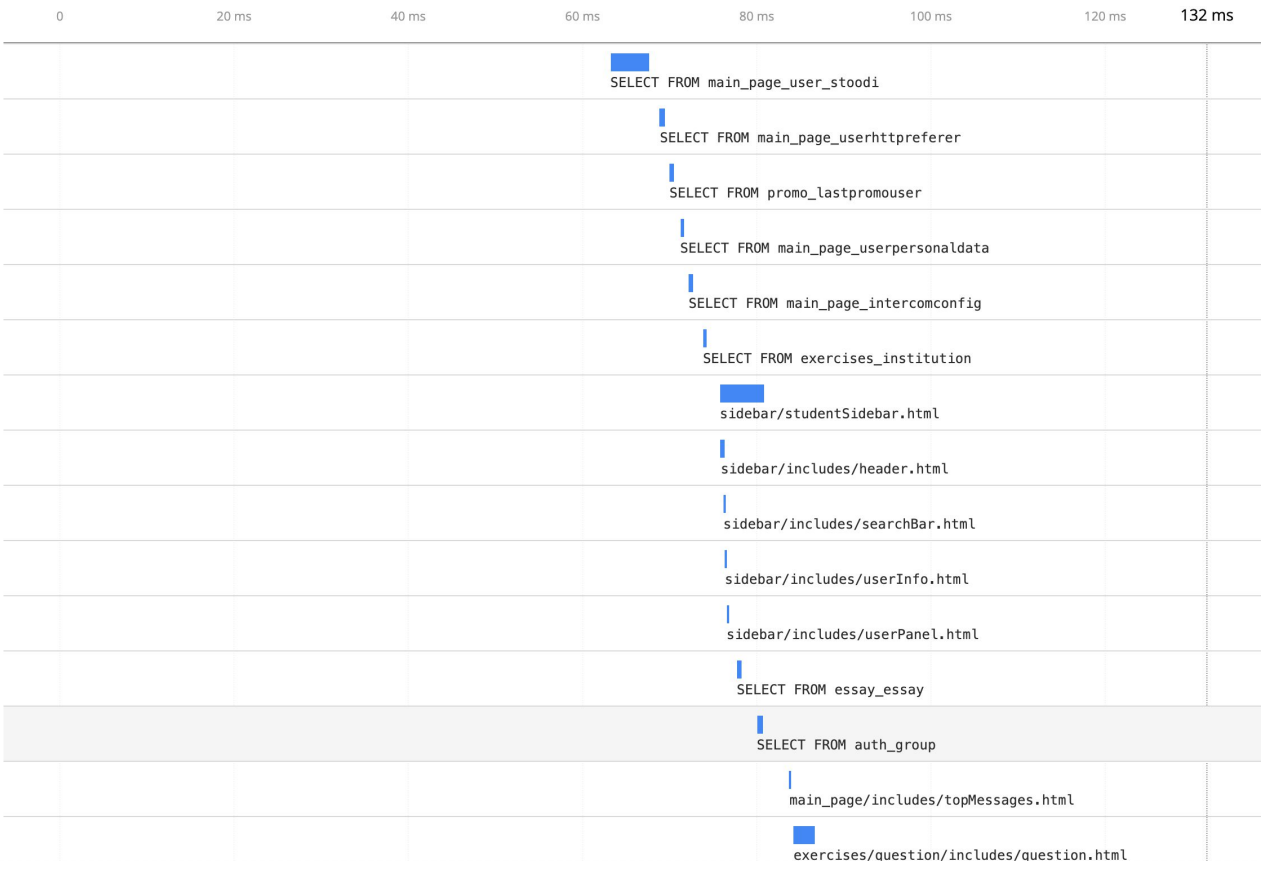
Kibana - Timeline



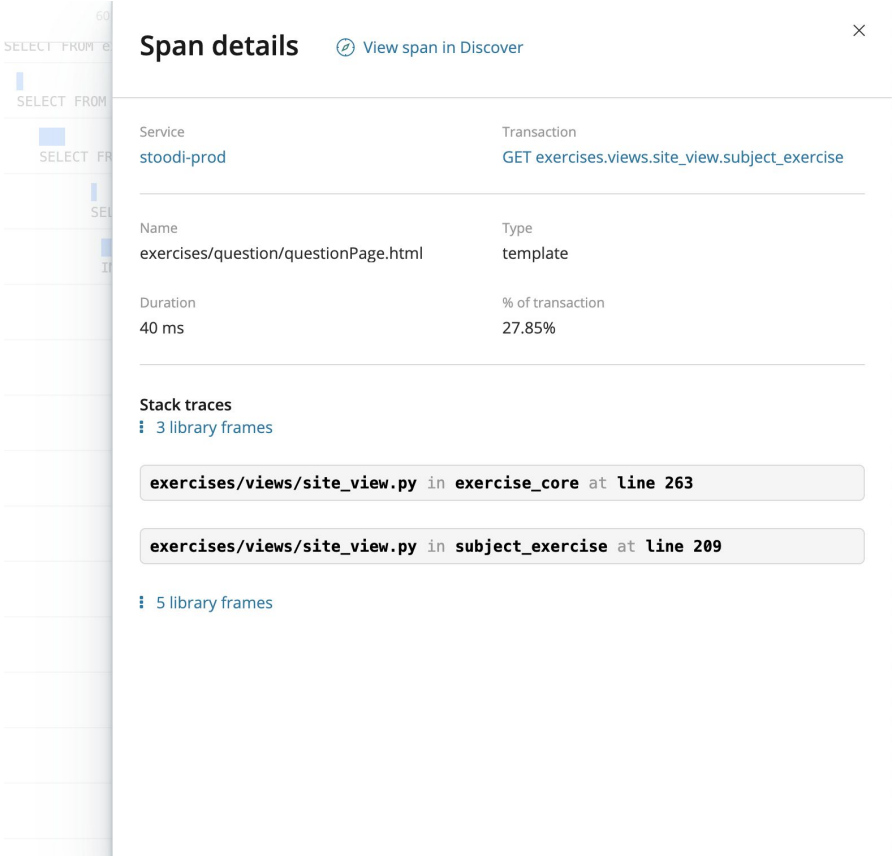
Kibana - Timeline



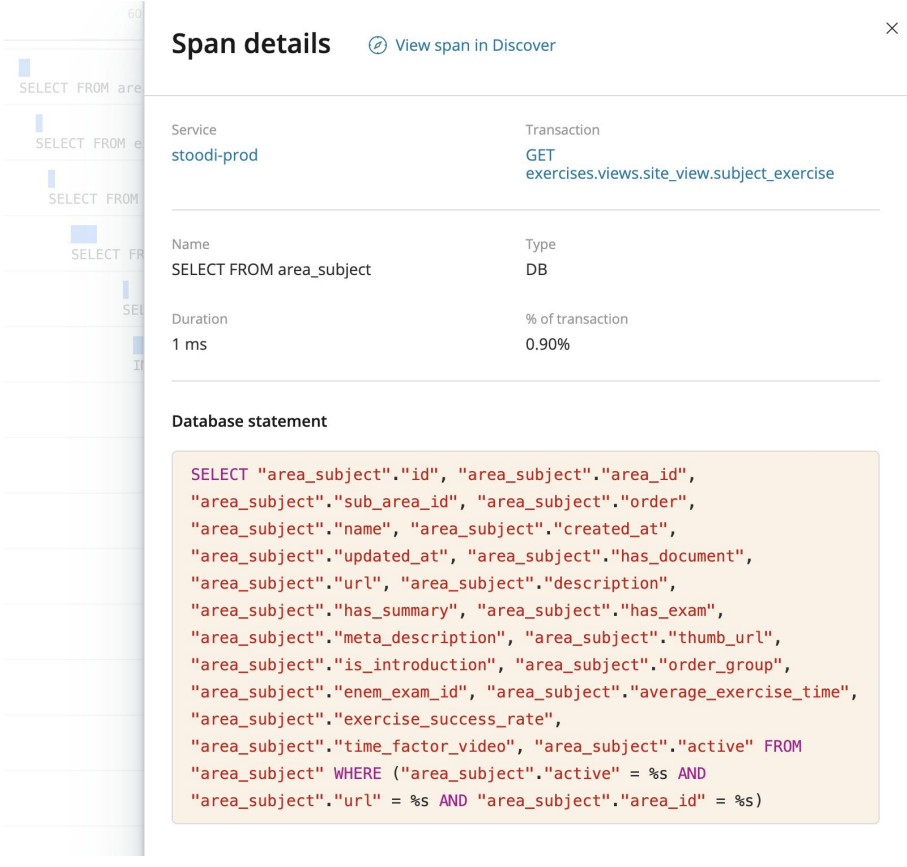
Kibana - Timeline



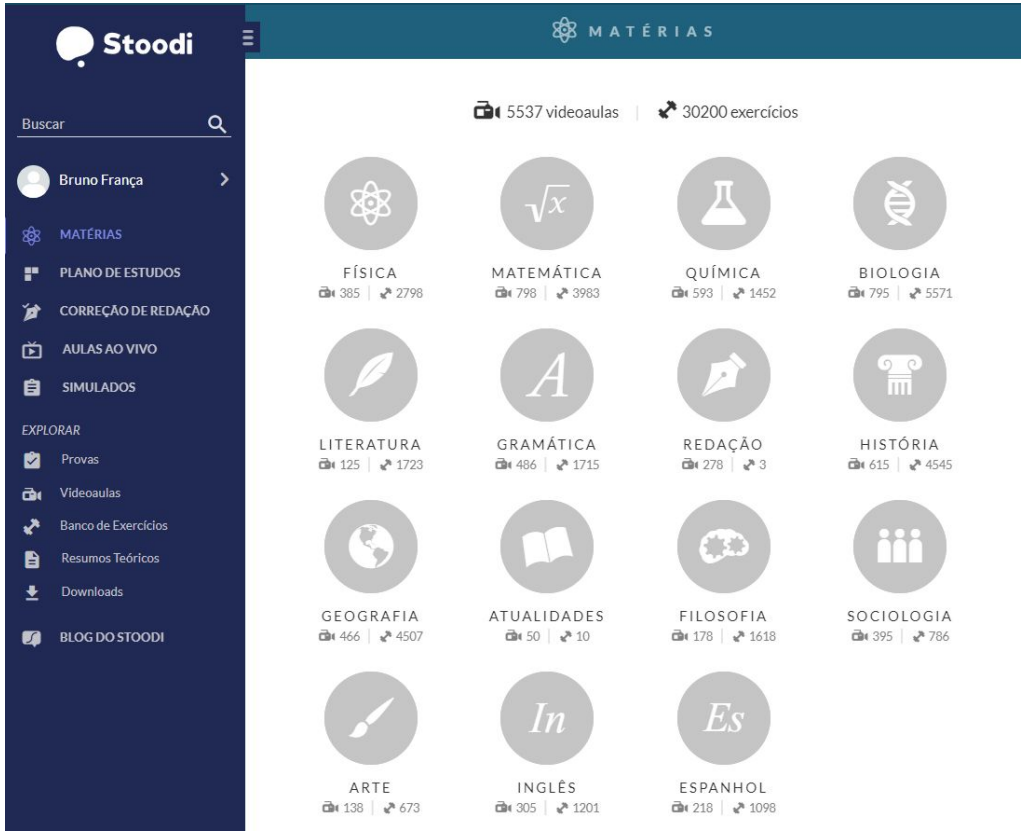
Kibana - Span Details



Kibana - Span Details



Exemplos


Stoodi
Buscar
Bruno França
MATÉRIAS
PLANO DE ESTUDOS
CORREÇÃO DE REDAÇÃO
AULAS AO VIVO
SIMULADOS
EXPLORAR
Provas
Videoaulas
Banco de Exercícios
Resumos Teóricos
Downloads
BLOG DO STOODI
MATÉRIAS
5537 videoaulas
30200 exercícios
FÍSICA 385 2798
MATEMÁTICA 798 3983
QUÍMICA 593 1452
BIOLOGIA 795 5571
LITERATURA 125 1723
GRAMÁTICA 486 1715
REDAÇÃO 278 3
HISTÓRIA 615 4545
GEOGRAFIA 466 4507
ATUALIDADES 50 10
FILOSOFIA 178 1618
SOCIOLOGIA 395 786
ARTE 138 673
INGLÊS 305 1201
ESPANHOL 218 1098

Exemplos

Antes

GET exercises.views.site_view.subject_exercise_pdf_list	3,826 ms	11,359 ms	3.6 rpm
GET summary.views.siteView.DownloadSummaryPDF	6,992 ms	43,559 ms	2.7 rpm

Depois

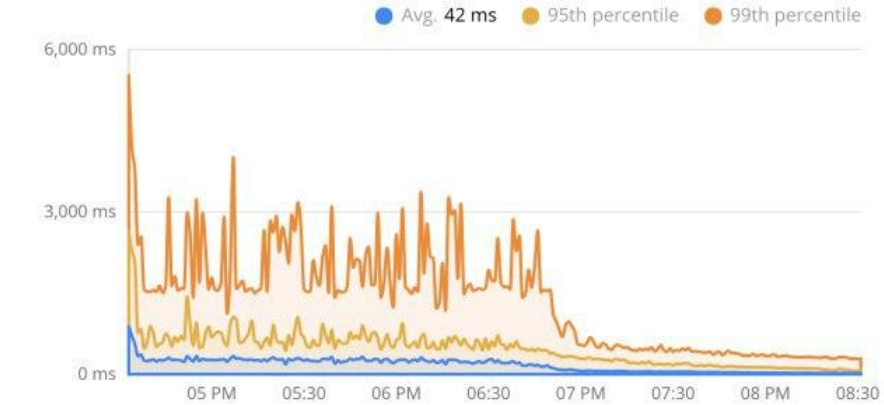
GET exercises.views.site_view.subject_exercise_pdf_list	1,098 ms	8,028 ms	1.8 rpm
GET summary.views.siteView.DownloadSummaryPDF	139 ms	199 ms	1.4 rpm

Exemplos

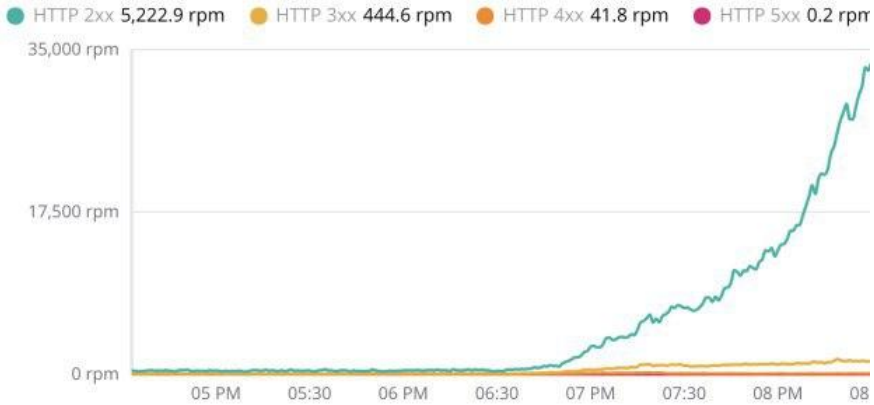
stoodi-prod

Request | Errors

Response times



Requests per minute



Request

O que preciso?

Elasticsearch

https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started-install.html

Kibana

https://www.elastic.co/guide/en/kibana/4.6/setup.html

APM

https://www.elastic.co/guide/en/apm/get-started/6.5/install-and-run.html

Como faço no meu projeto Django?

```
$ pip install elastic-apm
```

```
INSTALLED_APPS = (
    # ...
    'elasticapm.contrib.django',
)

ELASTIC_APM = {
    'SERVICE_NAME': '<SERVICE-NAME>',
    'SECRET_TOKEN': '<SECRET-TOKEN>',
}

MIDDLEWARE = (
     'elasticapm.contrib.django.middleware.TracingMiddleware',
     # ...
)
```

E no lambda?

https://github.com/UnitedIncome/serverless-python-requirements

```
$ pip install elastic-apm
```

E no lambda?

```python
import functools
import elasticapm

name = 'serverless_apm'


def get_apm_client():
    elasticapm.instrument()
    return elasticapm.Client()
```

```python
def apm_report(func):
    @functools.wraps(func)
    def execute(event, context):
        apm = get_apm_client()
        apm.begin_transaction('Request')
        elasticapm.set_custom_context({'event': event})

        try:
            result = func(event, context)
        except Exception:
            apm.capture_exception()
            raise
        finally:
            apm.end_transaction(func.__name__)
            apm.close()
            elasticapm.uninstrument()

        return result

    return execute
```

E no lambda?

```python
from serverless_apm import apm_report


@apm_report
def index(event, context):
    body = {
        'message': 'Olá mundo!'
    }

    return {
        'body': json.dumps(body),
        'statusCode': status_code,
        'headers': default_headers,
    }
```

Fique Atento!

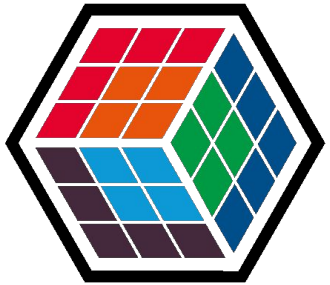
Auto Scale - Escalamos nossas instâncias de APM e Elasticsearch por quantidade de requests

Rotação de logs - Fazemos a nossa a cada ~ 4 meses

Quantidade de dados gerados - 60 Gb de dados por dia

É isso...

Obrigado!



THE DEVELOPER'S CONFERENCE