



DATADOG

APM Monitoring

Datadog In-depth Session (Day 2)
2:00 PM에 시작합니다!

SKT 2 days 교육 일정

시간	Day1
14:00 ~ 14:50 (정영석 부장)	Datadog Overview <ul style="list-style-type: none">- 기본 소개 및 전체 제품 구성 안내- 계정 접속 및 ORG 구성/권한- 기본 비용 체계- Q&A Infrastructure monitoring <ul style="list-style-type: none">- 개요 및 설명
14:50 ~ 15:00	Break
15:00 ~ 15:50 (정영석 부장)	Infrastructure monitoring <ul style="list-style-type: none">- 설치 및 구성 방법- 최적화 사용 방안- Q&A
15:50 ~ 16:00	Break
16:00 ~ 17:00 (이성욱 부장)	Log Management <ul style="list-style-type: none">- 설치 및 구성 방법- 최적화 사용 방안- Q&A

시간	Day2
14:00 ~ 14:50 (이성욱 부장)	Application Performance Monitoring (Back End) <ul style="list-style-type: none">- 개요 및 설명- 설치 및 구성 방법- 최적화 사용 방안- Q&A
14:50 ~ 15:00	Break
15:00 ~ 15:50 (정영석 부장)	Real User Monitoring (Front End) <ul style="list-style-type: none">- 개요 및 설명- 설치 및 구성 방법- 최적화 사용 방안
15:50 ~ 16:00	Break
16:00 ~ 17:00	Q&A Session

#TOPIC (APM)

1. Datadog APM Flow

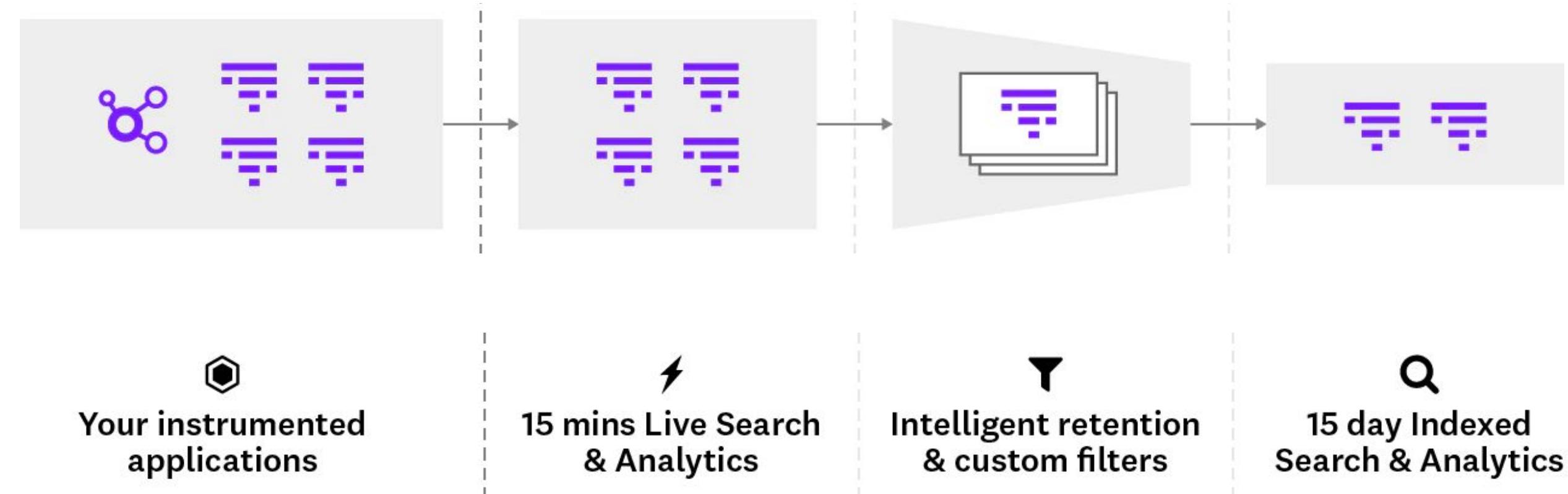
2. APM Features

3. APM Optimization

4. Q&A

APM Flow

#Datadog APM Flow



APM Features

#Sending Traces to Datadog

다양한 방법을 통한 Trace전송



Things To Check

- 버전 및 호환성 체크
- Tag 및 Option 설정
- Sampling/Filter Traces

Sample

```
export JAVA_OPTS="$JAVA_OPTS  
-javaagent:/app/datadog-trace/dd-java-latest.jar  
-Ddd.service.name=서비스명 -Ddd.env=환경  
-Ddd.version=버전 ...  
-Ddd.trace.header.tags=User-Agent:user-agent  
-Ddd.trace.db.client.split-by-instance=true  
-Ddd.http.server.tag.query-string=true ....
```

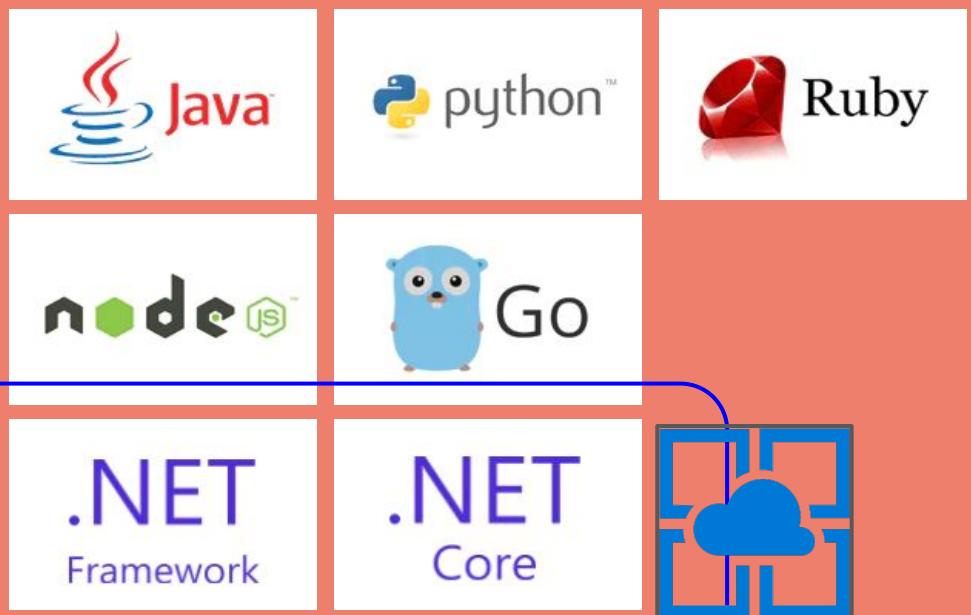
apm_config :

```
ignore_resources: ["GET /healthcheck"]
```



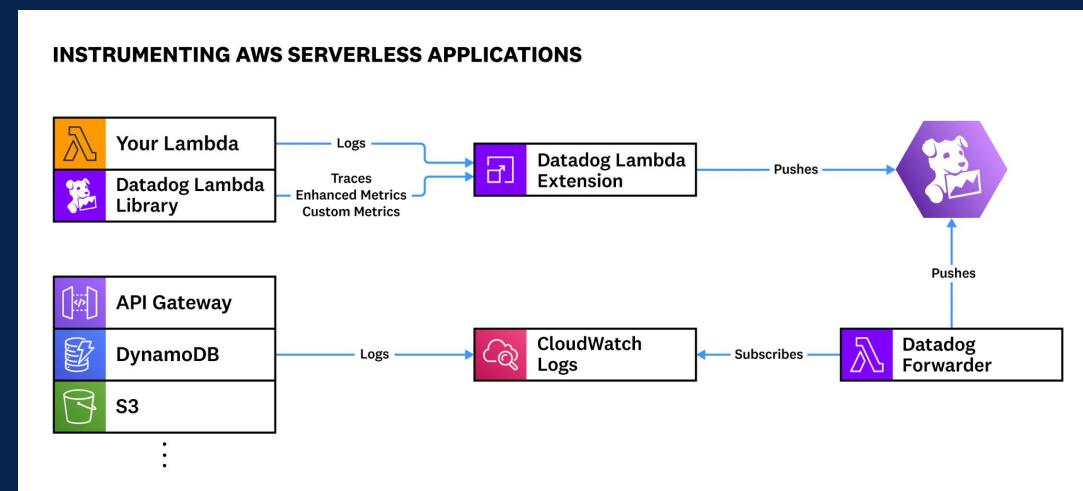
#Sending Traces to Datadog

Serverless



Things To Check

- 버전 및 호환성 체크
- Extension 추가



#Ingestion & Retention

Ingestion

- Ingestion Rate

Ex. DD_TRACE_SAMPLE_RATE=1.0

- Ingestion Status

Green - Complete

Gray - Filtered

Orange - tracer rate limiter (100 traces per second)

Red - dropped agent CPU limit (max_cpu_percent)



Retention

- 필터 조건에 따라 15일간 저장

Add a Retention Filter

- Name the filter
- Set filters for the span to match
- Set span retention rate

38.1M Matching spans in the past 15 minutes

DATE	SERVICE	RESOURCE	DURATION	HTTP METH...	STATUS COD
Aug 10 21:15:44.946	coffee-house	RecordOrder	40.43ms		200
Aug 10 21:15:44.923	ad-server	GET /find_ad/parameters.json	1994.14μs	GET	200
Aug 10 21:15:44.517	send-email-redis-queue	PUBLISH	955.59μs		
Aug 10 21:15:44.362	product-recommendation	GET /report	567.31ms	GET	200
Aug 10 21:15:44.347	web-store	ShoppingCartController#apply_-	649.66ms	POST	200
Aug 10 21:15:44.223	php.frontend.site	CoffeeHouse\Controllers\HomeC...	4.21s	GET	200
Aug 10 21:15:43.970	send-email-redis-queue	PUBLISH	775.78μs		
Aug 10 21:15:43.878	web-store	PUT 500	15.09s	PUT	500
Aug 10 21:15:43.683	web-store	ApplicationController#health	24.42ms	GET	200
Aug 10 21:15:43.599	shipping-queue-redis	BRPOP	1233.55ms		

Retain 100 % of matching indexed spans
These spans will be stored for 15 days for search and analytics with the full trace flamegraph when available.

#Additional Configuration

추가 설정

- Logs Injection - trace_id and span_id
- Unified Service Tagging - version, env, and service
- Collect runtime metrics



Benefit

- APM/LOG 연계 분석
- 배포 전후 분석 및 환경별 분석
- Runtime Metric 비교 분석

```
<appender name="FILE" class="org.apache.log4j.FileAppender">
    <param name="File" value="logs/app.log"/>
    <param name="Append" value="true"/>

    <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %-5p
Log4j
%c{1}:%L - %X{dd.trace_id} %X{dd.span_id} - %m%n"/>
    </layout>
</appender>
```

#Common Inquiry

FAQ

- Deployments 항목이 비활성화되어 있습니다
- 두개의 DB가 하나로 표시됩니다
- 특정 Request Header값을 추적하고 싶습니다
- /healthcheck 요청을 제외하고 싶어요
- Dashboard에서 View related traces 관련 요청을 볼수 없습니다

Things To Check

Sample

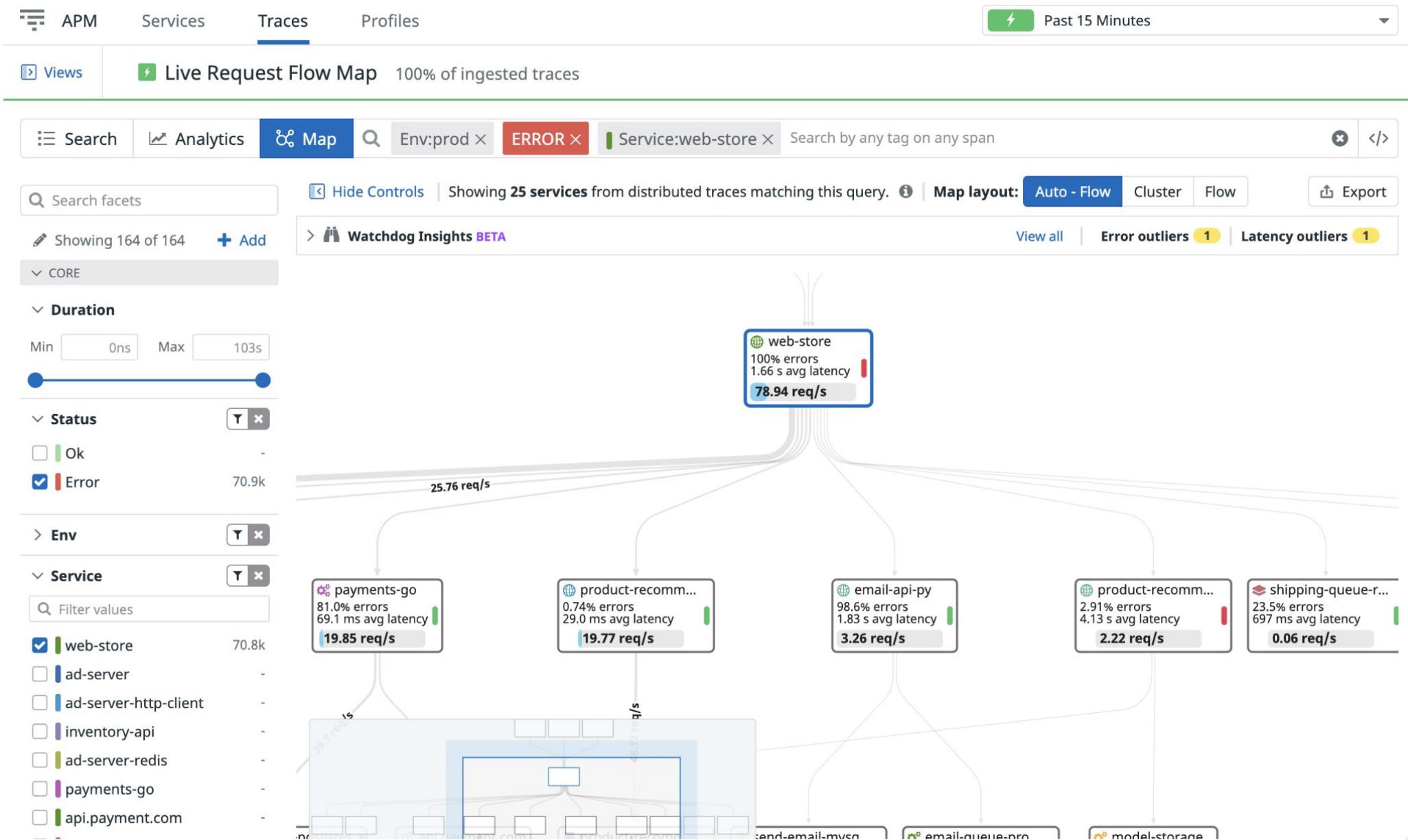
```
export JAVA_OPTS="$JAVA_OPTS  
-javaagent:/app/datadog-trace/dd-java-latest.jar  
-Ddd.service.name=서비스명 -Ddd.env=환경  
-Ddd.version=버전 ...  
-Ddd.trace.header.tags=User-Agent:user-agent  
-Ddd.trace.db.client.split-by-instance=true  
-Ddd.http.server.tag.query-string=true ....
```

apm_config :

```
ignore_resources: ["GET /healthcheck"]
```



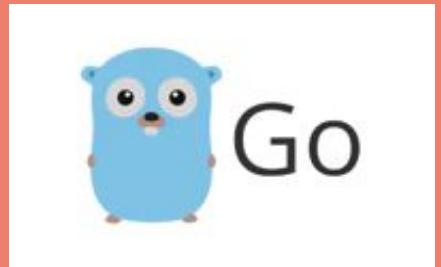
#APM Features Demo



Profiler & Database Monitoring

#Profiler

지원 언어



Things To Check

- 지원 버전 Check

Ex.

OpenJDK 11+, Oracle JDK 11+,
OpenJDK 8 (version 8u262+)



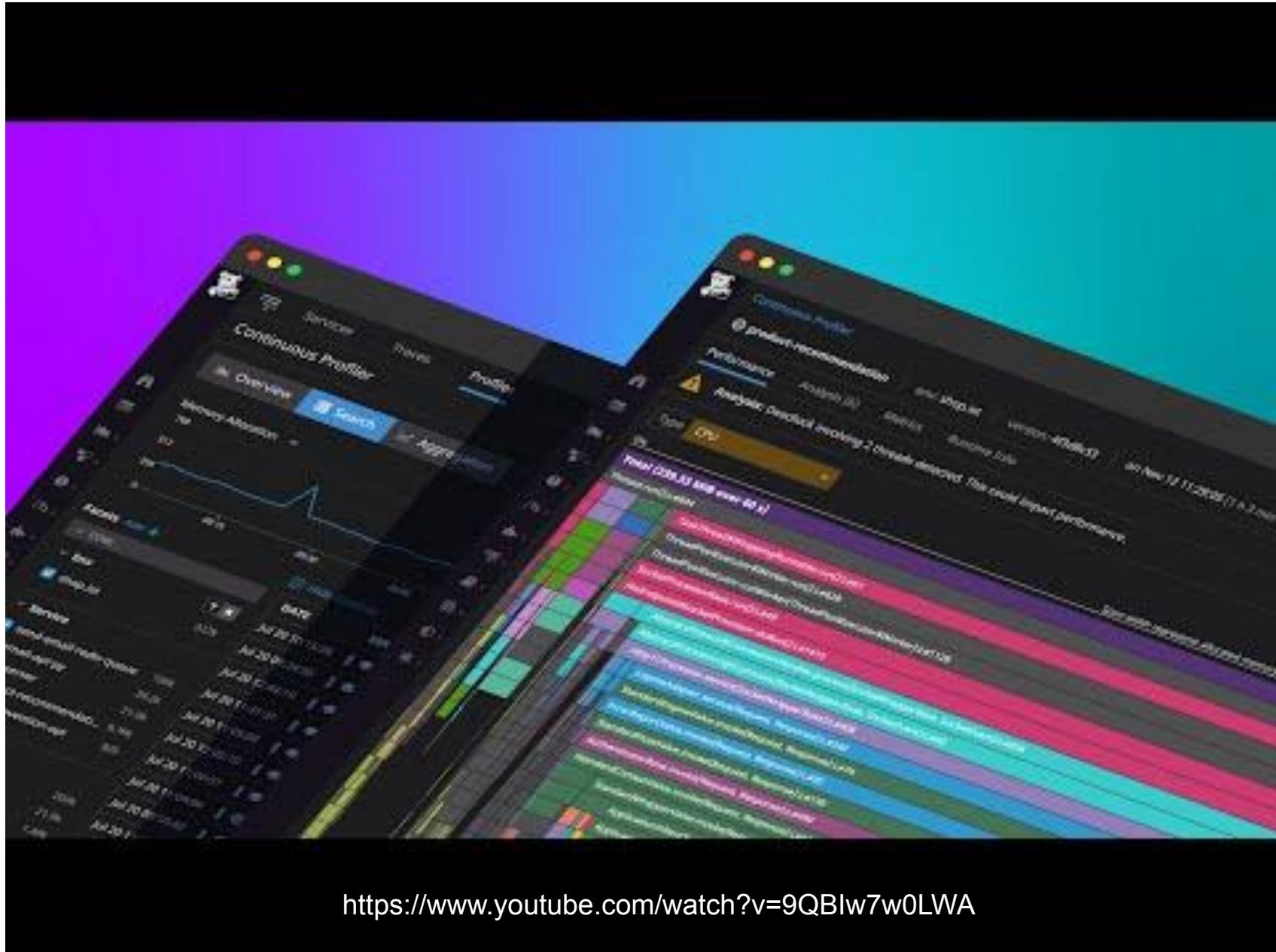
#Profiler

APM & Profiler 비교

	APM & Distributed Tracing	Profiling
Visibility level	Services & 3rd party libraries	100% of the code (methods, classes, threads)
Measurement	Requests, Errors, Duration (latency)	Resource consumption (cpu, memory & time)
Flame graph	Execution path of a request across multiple services, including duration and errors	Aggregated 1-min view of resource consumption per method, per application process

Traces tell you **WHICH** requests were slow at the **service** level
Profiles tell you **WHY** they were slow at the **line of code** level

#Profiler



#Database

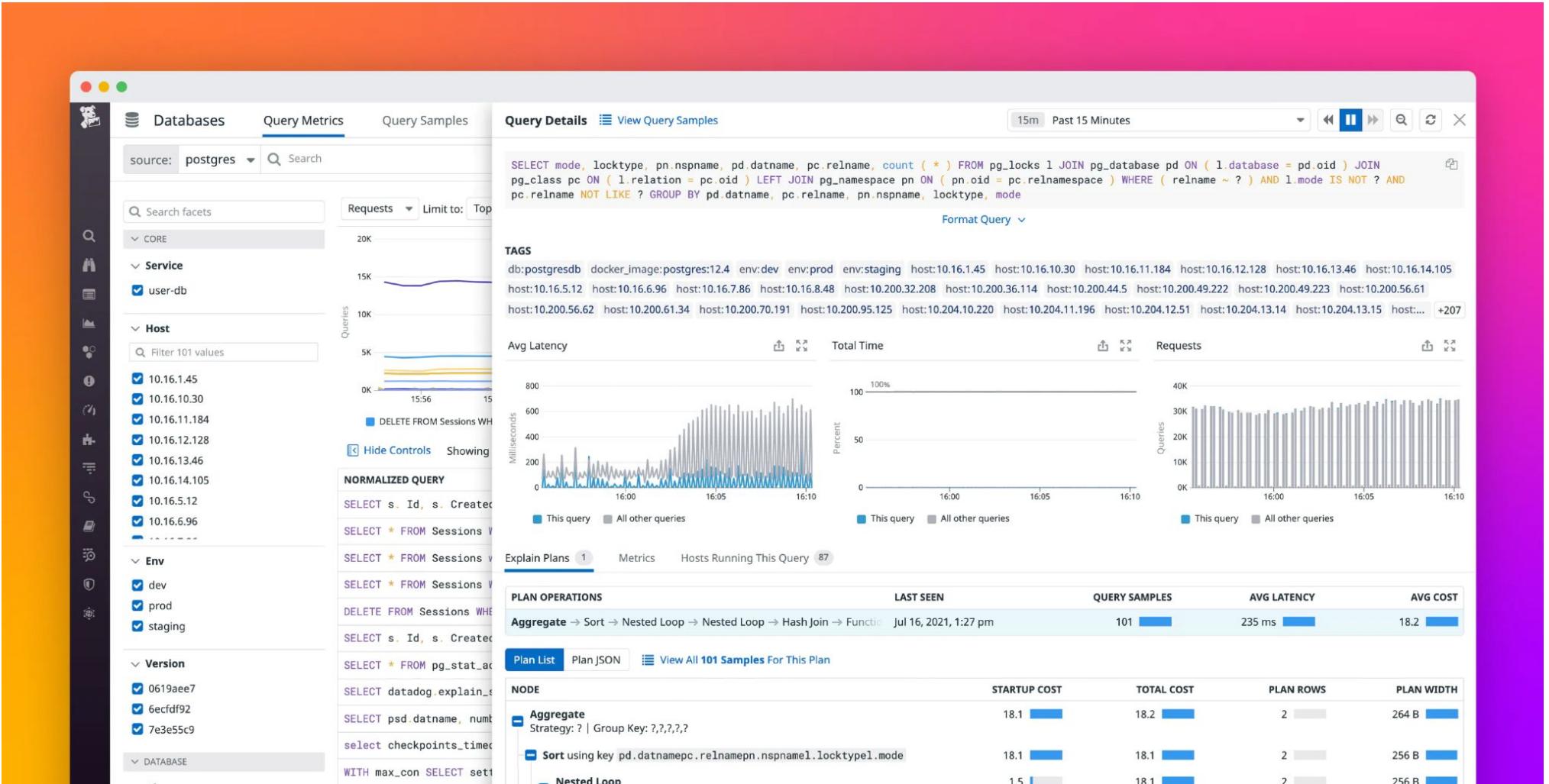
Database Monitoring

	APM & Distributed Tracing	Database
Visibility level	Services & 3rd party libraries Including Database Query Call	performance of normalized queries historical trends in query performance
Measurement	Requests, Errors, Duration (latency)	slow queries with detailed explain plans & costs
Flame graph (Details Provided)	Execution path of a request across multiple services, including duration and errors	Sampled queries with trend / query plans and costs

Traces tell you **WHICH** requests(query) were slow at the **service(resource)** level
Database tell you **WHY** they were slow based on **Trends, Plans & Costs** details

#Database

Database Monitoring



APM Optimization

#APM Optimization

Trace Without Limits™

- INGEST
 - Filter Logs
 - Generate Metrics
- INDEX
 - Trace 중요도별 저장 유무 조정
 - 사용량에 따른 알람 생성

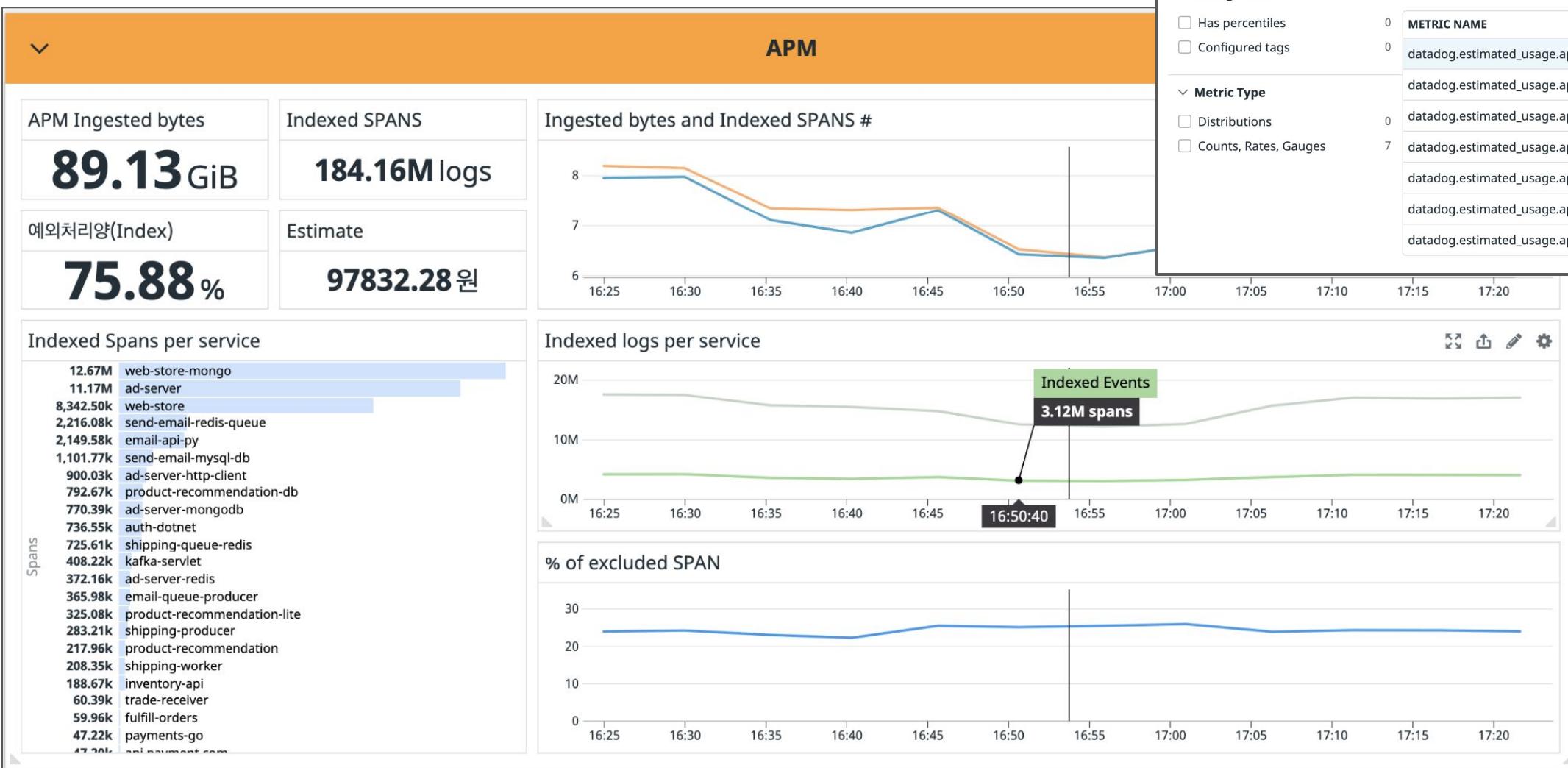
After

- INGEST
 - Ingest 비용 절감
 - Trace 저장 없이 특정 패턴 발생시 알람
- INDEX
 - Error 혹은 @duration: >1s Trace 저장
 - Dashboard 및 알람으로 특이사항 감지



#APM Optimization

Metric을 활용한 사용량 모니터링



#Quick Review

Q1. Code level 분석시 적합한 Datadog Solution은

1. APM
2. Continuous Profiler
3. Database Monitoring

Continuous Profiler

Q2. 배포 이후 전후 특이사항 분석을 위해 필요한 Tag는

1. DD_ENV
2. DD_SERVICE
3. DD_VERSION

DD_VERSION Tag



Real User Monitoring

오늘 살펴볼 내용..

□ 프론트 개발자 Mike의 Datadog 활용법

1. 사용자 요청의 병목 구간 찾기
2. 지역별 가용성 파악하기
3. 프론트엔드 Error Tracking
4. RUM Analytics

□ RUM Best Practice

□ 실 서비스에 RUM 맛보기

프론트 개발자 Mike의 Datadog 활용법

#1 사용자 요청의 병목 구간 찾기

프론트 개발자 Mike는 신규 페이지 구현 요청을 받고...

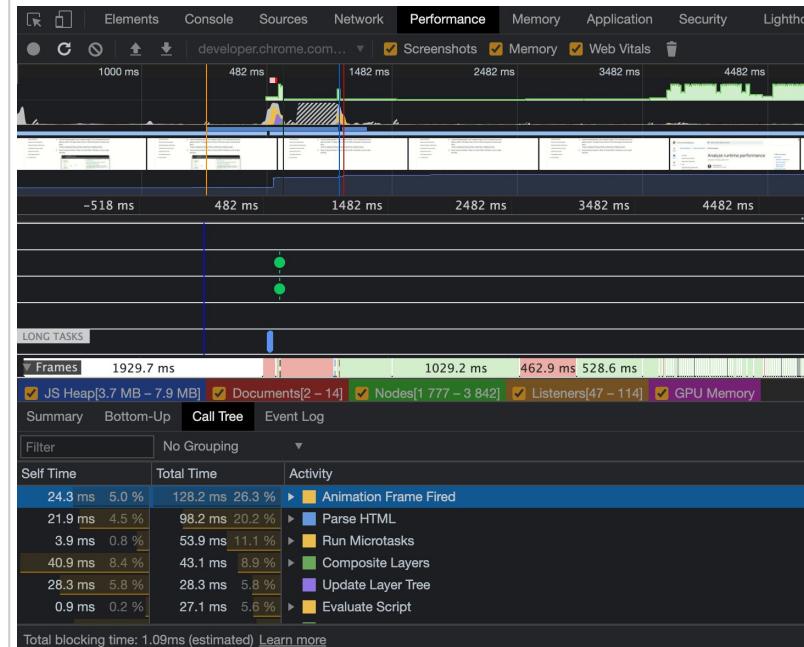
1. 기능개발

- Peer 코드 리뷰
- Unit test



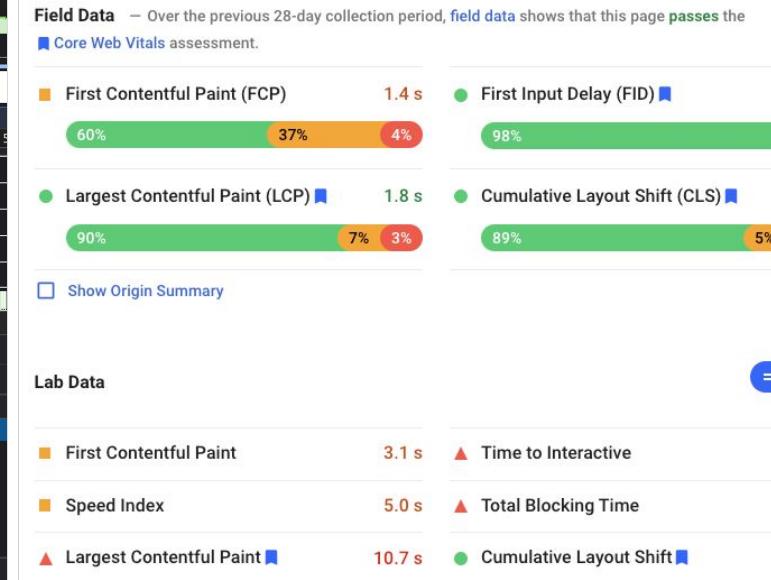
2. Performance 최적화

- 로딩 최적화
- 렌더링 최적화



3. Staging 테스트

- QA 팀 검증 완료
- Production 배포!



브라질 사용자들이 페이지 로딩이 너무 느리다는 리포트..

내 자리에서는 잘만
열리는데.. QA도 다
통과했는데.. 왜지?

Datadog으로 프론트엔드 성능
추적이 된다고 한거 같은데...
한번 적용해 봄야겠다

팀장님.. Blah Blah~



Datadog RUM을 연동하기로 결정

1. <https://www.datadoghq.com/>에서 Free Trial 가입

Get Started with Datadog ×

No credit card required
Try it free for 14 days and monitor as many servers as you like.

*Required Fields

Region*
Please choose carefully. You can't migrate data between regions.
Where do you want your data housed?

United States (US1) ▼

Email*

Full Name*

Company*

Password*
Use at least 8 characters containing at least 1 number and 1 lowercase letter

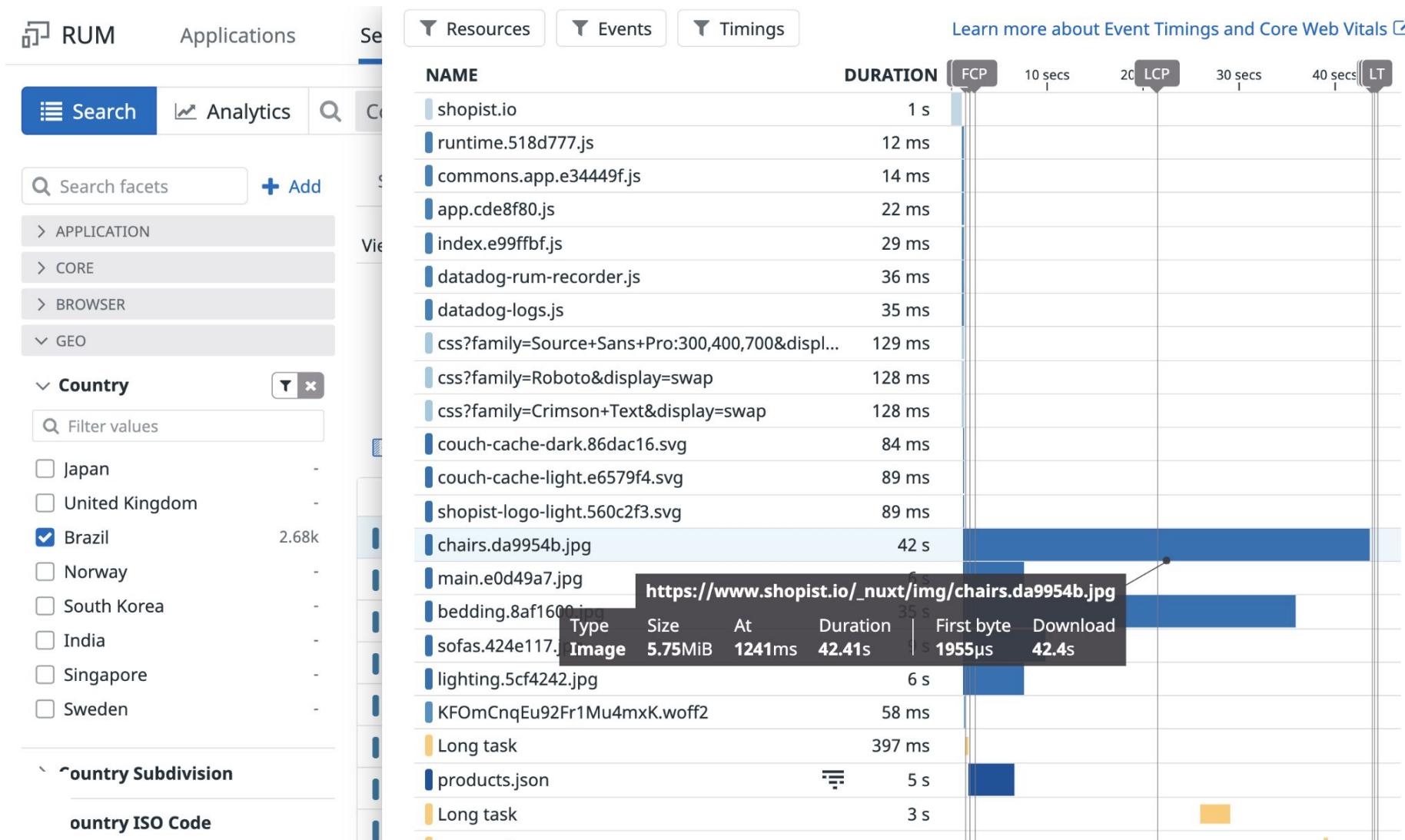
Phone

2. Datadog에서 제공하는 Code를 트래킹을 희망하는 페이지에 삽입

```
import { datadogRum } from '@datadog/browser-rum';

datadogRum.init({
    applicationId: 'c2039693-338b-*****-8864-50aed4e4b2c0',
    clientToken: 'pub6310d6*****a1e749fabd7e25',
    site: 'datadoghq.com',
    service:'event',
    env:'production',
    sampleRate: 100,
    trackInteractions: true
});
```

Datadog RUM을 이용한 병목 구간 확인 (Demo)



Mike가 Datadog을 통해 해결 할 수 있었던 것들..

1. 특정 지역의 사용자 성능 이슈를 바로 파악하여 해결 방안 제시
2. 프론트와 백엔드의 가시성을 확보하여 백엔드 팀과 원활한 이슈 분석

프론트 개발자 Mike의 Datadog 활용법

#2 지역별 가용성 파악하기 (Synthetics)

우리가 사용자 보다 먼저 이슈를 파악하고 싶어요..

브라질에 우리 상주인력도
없는데 어떻게 테스트를
하지..

Datadog Synthetics를
이용하면 지역별 가용성
파악이 가능하다고 한거
같은데..



우리가 사용자 보다 먼저 이슈를 파악하고 싶어요..[Demo](#)

1. Synthetic에서 테스트할 URL 및 지역을 설정

1 Set your test details

Starting URL
https://shopist.io/event

> Advanced Options

Name
이벤트 페이지 측정

Environment (env)
env:prd X

Additional Tags
Select or add related tags

Browsers & Devices

	Chrome	Firefox BETA
Laptop Large	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tablet	<input type="checkbox"/>	<input type="checkbox"/>
Mobile Small	<input type="checkbox"/>	<input type="checkbox"/>

2 Select locations

All Locations (15)

Managed Locations (15)

Americas (5)	Asia Pacific (5)	Europe (5)
Canada Central (AWS)	Mumbai (AWS)	Frankfurt (AWS)
N. California (AWS)	Seoul (AWS)	Ireland (AWS)
Ohio (AWS)	Singapore (AWS)	London (AWS)
Oregon (AWS)	Sydney (AWS)	Paris (AWS)
São Paulo (AWS)	Tokyo (AWS)	Stockholm (AWS)

2. 테스트 희망하는 경로를 레코딩

Synthetics > Test details > Edit Browser Test > Edit Recording

이벤트 페이지 측정
Run from 15 locations and 2 devices

Stop Recording

Add New:
✓ Assertion ↗ Navigation + Special Actions
+ Variables Subtest ❤ HTTP Request

9 steps recorded Clear All Steps

- 1 Click on div "Sofas"
- 2 Click on image "12.90a1baf.jpg"
- 3 Click on div "Add to cart"
- 4 Click on div
- 5 Click on div "Cart (1)"
- 6 Click on div "Checkout"
- 7 Click on div "Continue shopping"
- 8 Click on div "hamburger-item"
- 9 Click on div "Lighting"

COUCH CACHE

FURNITURE

Edison Hanging Bulbs
\$250.00



Chrome Arc Lamp
\$320.00



Geometric Cage Hanging Lamp
\$210.00



Laser Cut Hanging Lamp
\$210.00

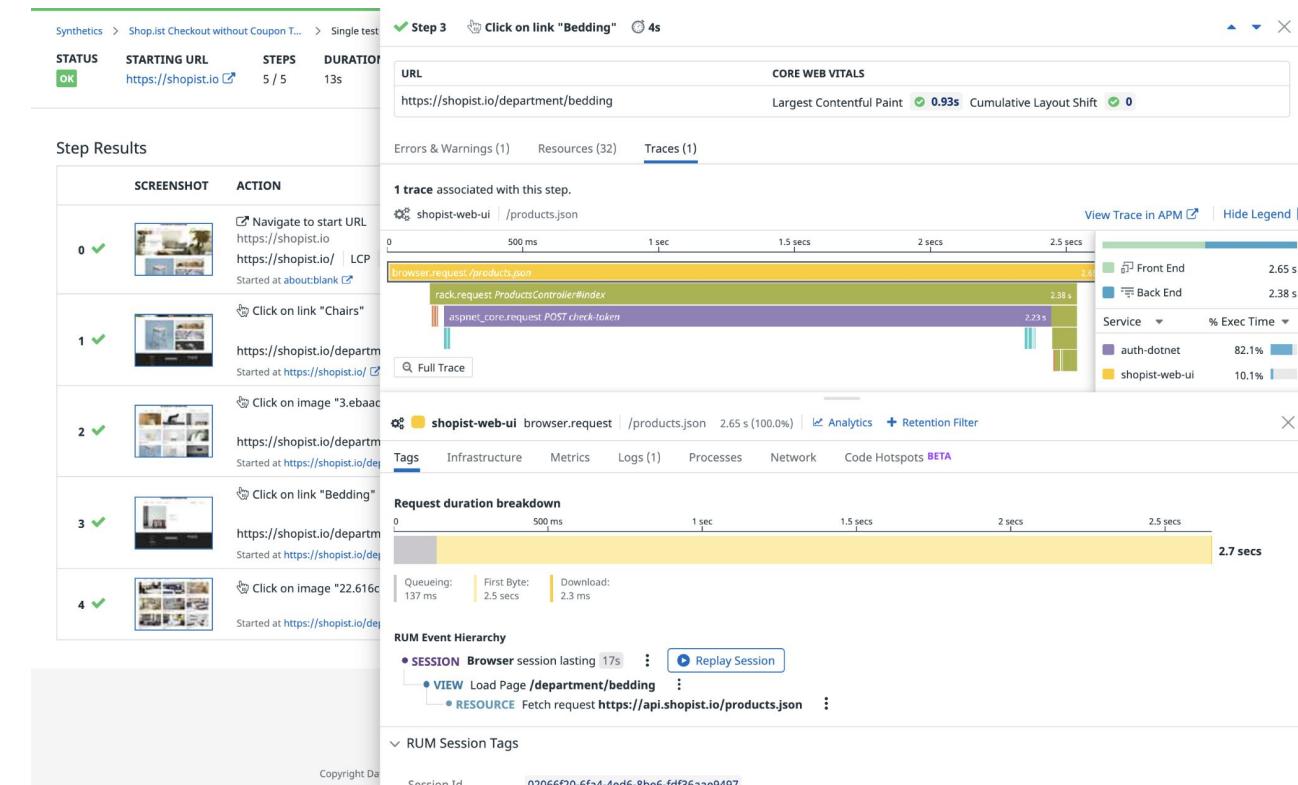


Mike가 Synthetics 통해 해결 할 수 있었던 것들..

1. 각 지역별 성능 및 가용성 확인



2. 문제 발생시 백엔드와 연계되어 신속하게 이슈 분석



프론트 개발자 Mike의 Datadog 활용법

#3 프론트엔드 Error Tracking

사용자 환경에서 발생하는 에러를 수집해 주세요..

응답 결과나 에러 로그를
버퍼에 모았다가 서버로
전송을 해야하나?

로그를 모아서 관리는
어떻게 하지?

Datadog에서 에러도 트래킹
된다고 한거 같은데..



[Demo](#)

RUM Error Tracking을 통해 해결 할 수 있었던 것들..

1. 사용자에게 발생하고 있는 에러에 대해 수집
2. 발생한 에러가 어떤 컨텍스트에서 발생했는지 연계 분석
3. 새로운 유형의 에러가 발생하면 알람을 받을 수 있도록 조치

프론트 개발자 Mike의 Datadog 활용법

#4 RUM Analytics

이번 이벤트에 유입된 사용자의 통계를 보고 싶어요..

사용자 이벤트 로그를
수집해야 되나..

백엔드 담당자가 DB로
주요 지표는 남겨놨을텐데..

Datadog에서 제공하는 RUM
Analytics를 활용해 봐야겠다



다음의 사용자 지표에 대해 파악을 희망

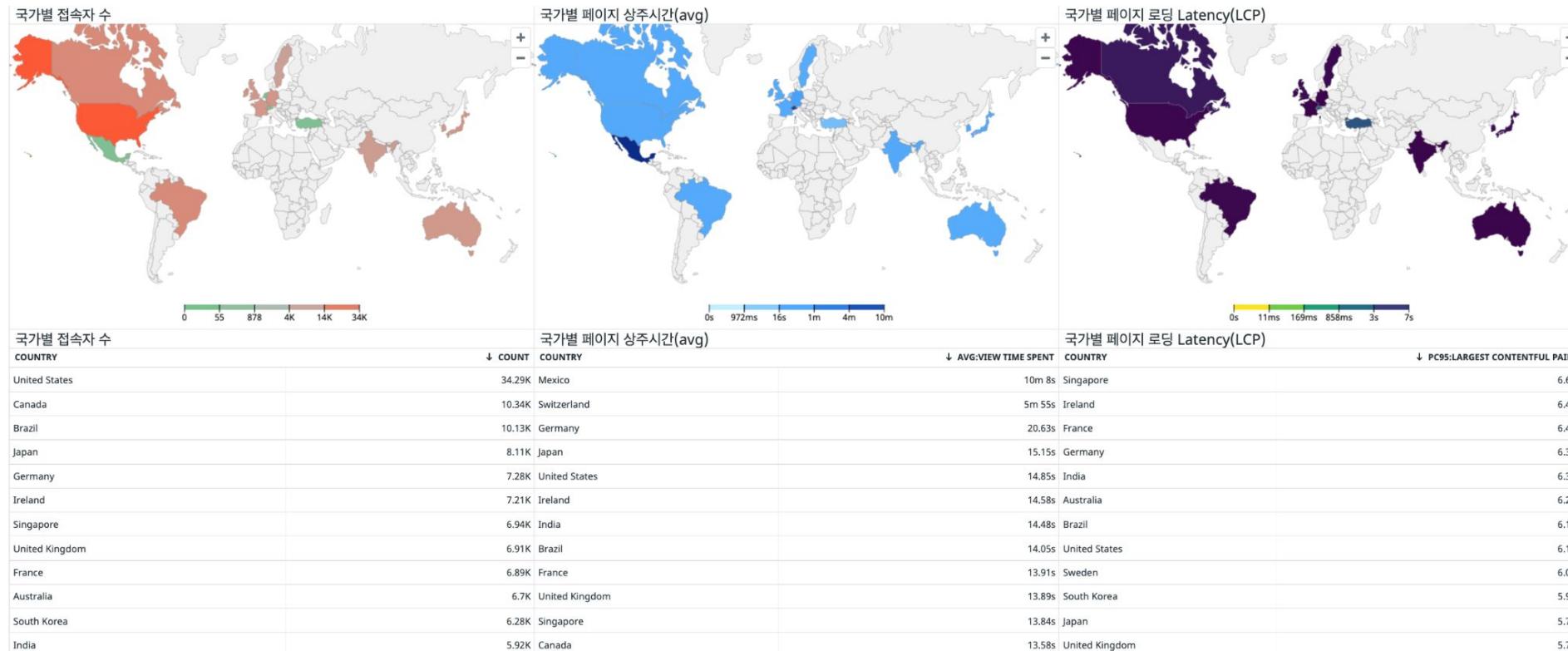
1. 국가별 접속자 수를 알고 싶어요
2. 페이지별로 사용자가 얼마나 시간을 보냈는지 알고 싶어요
3. 국가별로 페이지 로딩 시간 지표를 확인하고 싶어요

[Demo](#)

RUM Analytics을 통해 해결 할 수 있었던 것들..

1. 수집된 사용자 정보 기반으로 통계 정보 확인

2. 통계 정보를 Dashboard로 Export하여 간단하게 이벤트 현황판 구성



그외의 RUM 주요 기능

Session replay

End User의 WebSite 여정을 Replay하여 어떤 부분에 문제가 있었는지 쉽게 파악할 수 있도록 기능 제공

This screenshot captures a browser session replay interface, likely from a tool like Session Replay or similar. The main area shows a user navigating through a lighting department on a website. The top navigation bar includes links for Sessions, Browser session replay lasting 38s by user Britt Doe, shopist-web-ui env:prod version:5cb64d0 Chrome Linux United States, IN PROGRESS, Share, and Event Timeline.

The main content area displays a grid of lighting products:

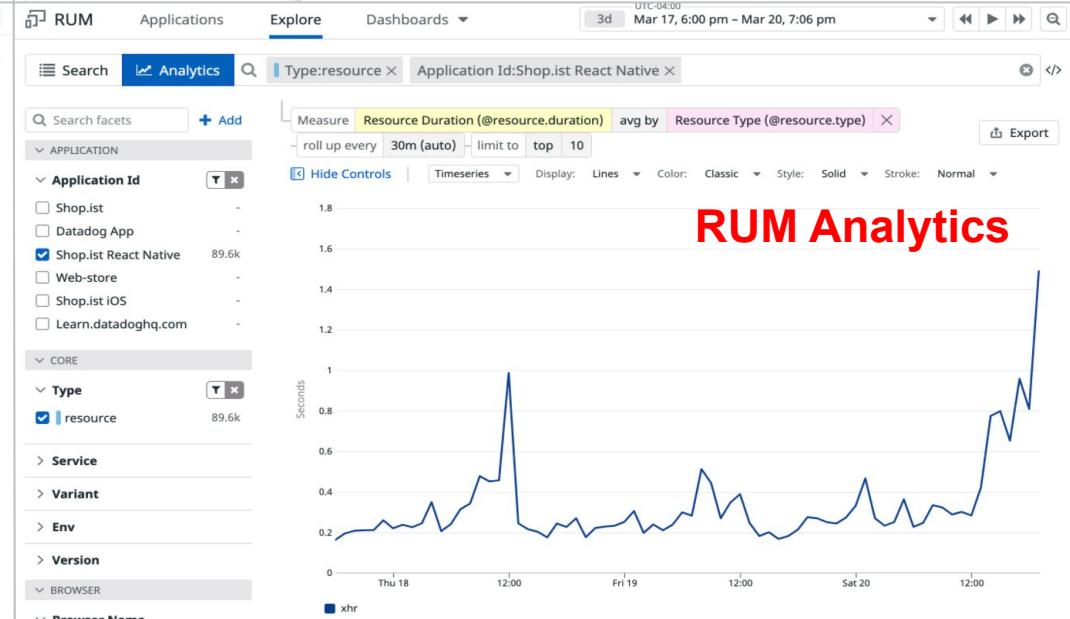
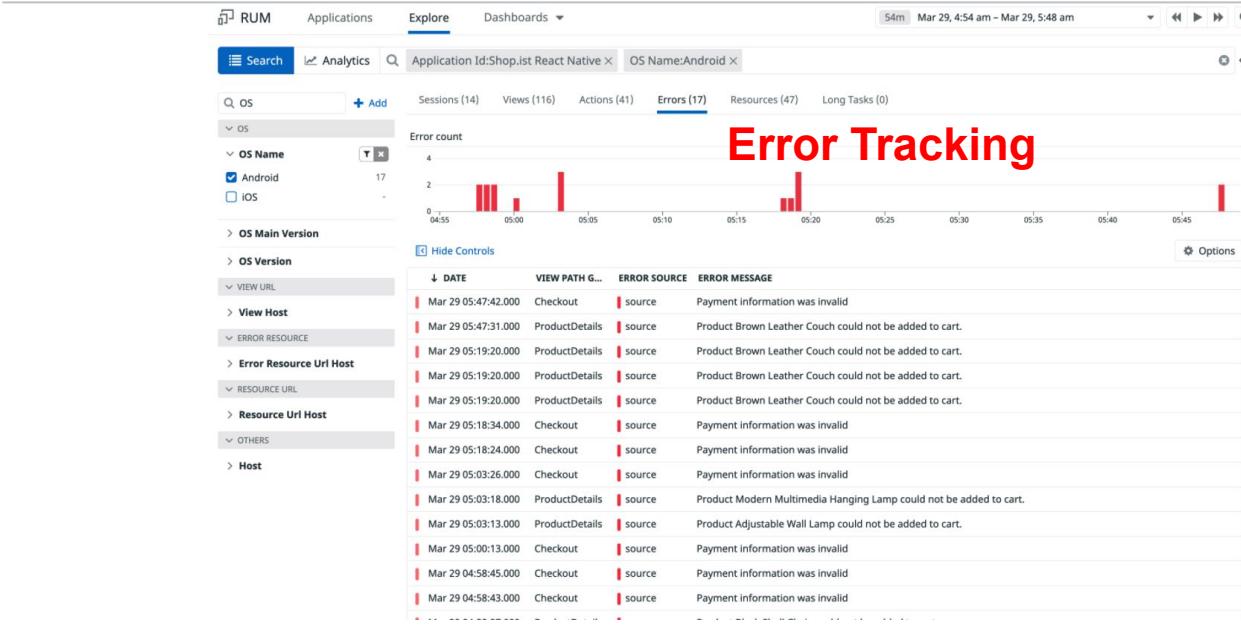
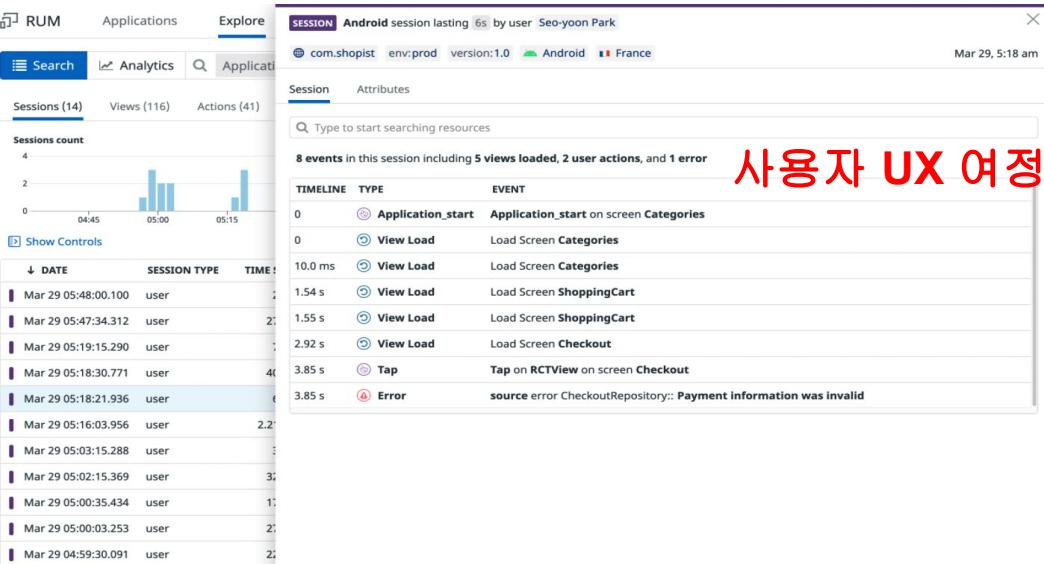
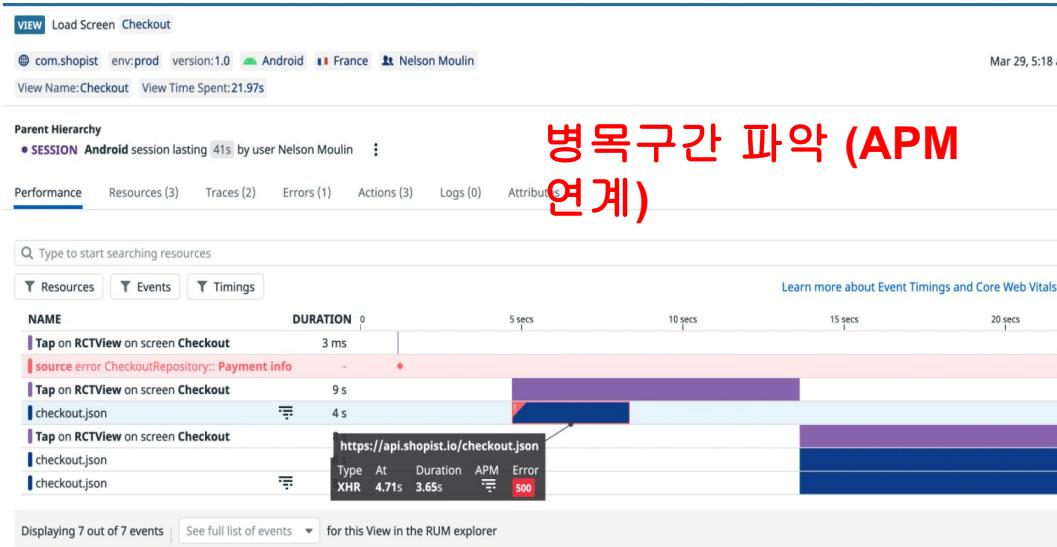
- Edison Hanging Bulbs**: \$250.00, Sold Out
- Chrome Arc Lamp**: \$320.00, In Stock
- Geometric Cage Hanging Lamp**: \$210.00, In Stock
- Laser Cut Hanging Lamp**: \$210.00, In Stock
- Matte Grey Floor Lamp**: \$210.00, In Stock
- Vintage Desk Lamp**: \$210.00, In Stock
- Glass Hanging Lamp**: \$210.00, In Stock
- Modern Multimedia Hanging Lamp**: \$210.00, In Stock
- Adjustable Wall Lamp**: \$210.00, In Stock

The bottom of the screen features a timeline bar indicating the session duration (00:20 / 00:38) and playback controls (Skip inactivity, .25x, .5x, 1x, 2x, 4x).

The right side of the interface shows the **EVENT TIMELINE**, listing the sequence of events:

- 0 Load Page /
- 19.6 s Load Page /department/lighting
- 20.6 s Load Page /department/lighting/product/?
- 22.6 s Click on ADD TO CART on page /department/lighting/product/?
- 22.6 s source error TypeError: Cannot read property 'addUserAction' ...
- 22.6 s console error console error: TypeError: Cannot read property '...' ...
- 30.7 s Load Page /profile
- 34.9 s SPA Route Change /profile-edit
- 36.6 s Click on SAVE PROFILE on page /profile-edit
- 36.6 s source error TypeError: Converting circular structure to JSON -...
- 36.6 s source error TypeError: Converting circular structure to JSON -...
- 36.6 s console error console error: TypeError: Converting circular str...
- 37.5 s source error ReferenceError: profile is not defined
- 37.5 s source error ReferenceError: profile is not defined
- 37.5 s console error console error: ReferenceError: profile is not defined
- 37.5 s SPA Route Change /profile

React Native Support



Mobile RUM (Android/iOS) Crash report

Crash Error에 대해 트래킹

ISSUE  io.shopist.android |  java.lang.RuntimeException: Application crash detected

Total errors: 4.68k | First seen: 2 months ago | version:1.3.4 | Last seen: about 1 hour ago | version:1.3.9

14 days



Latest available errors

ERROR Apr 29 14:08:53.789 (about 1 hour ago)

version:1.3.9 env:prod

 **java.lang.RuntimeException:**
Application crash detected

```
at io.shopist.android.data.source.DefaultCategoryRepository.refreshCategories(DefaultCategoryRepository.kt:25)
at io.shopist.android.home.HomeViewModel$$special$$inlined$switchMap$2$lambda$1.invokeSuspend(HomeViewModel.kt:33)
at kotlin.coroutines.jvm.internal.BaseContinuationImpl.resumeWith(ContinuationImpl.kt:33)
```

Show all 70 frames

```
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:492)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:930)
```

View

View ID: 59bdcaa24-a8ad-40b8-a54c-8f86aeb86b40

View URL

Session

Session Id: 15d92332-54ed-4b9d-ad7e-5d1c8c43c2de

GEO

Continent: Europe

어떤 UX 여정 중 Crash가 발생했는지 분석

ISSUE  io.shopist.android |  java.lang.RuntimeException: Application crash detected

Total errors: 4.68k | First seen: 2 months ago | version:1.3.4 | Last seen: about 1 hour ago | version:1.3.9

DEVICE

Device Type: Mobile

OS Name:  Android

OS Version: 10

SESSION TIMELINE

4 events in this session including 1 view loaded, 2 user actions, and 1 error

TIME	TYPE	EVENT
14:08:47	View Load	Load Screen io/shopist/android/home/HomeFragment
14:08:47	Application_start	Application_start on screen io/shopist/android/home/HomeFragment
14:08:52	Swipe	Swipe on RecyclerView(recycler) on screen io/shopist/android/home/HomeFragment
14:08:52	Error	source crash java.lang.RuntimeException: Application crash detected

View Entire Session in RUM

User

User Id: d4dbbe9b-e1da-4471-b9b9-e20b35c6f0d1

User Email: milos.lemmer@example.com

User Name: Milos Lemmer

IP Address: 92.160.101.137

Libraries

sdk_version: 1.9.0-alpha2

Custom Attributes

```
{
  "campaign": {
    "origin": "googleAds"
  },
  "usr": {
    "handle": "milos.lemmer@example.com",
    "type": "standard"
  }
}
```

프론트 개발자 Mike의 Datadog 활용법 요약

❖ Datadog RUM

- 사용자 관점의 성능 지표를 상세 분석
- APM 연동하여 백엔드 지표와 함께 연계된 분석
- 수집된 사용자 데이터를 통해 비지니스 인사이트 분석

❖ Datadog Synthetic

- 글로벌 테스팅 노드를 활용하여 우리 서비스의 지역별 성능 파악
- 문제 탐지 시 백엔드와 연계되어 신속하게 분석할 수 있도록 지원

RUM Best practice

Recommendations

1. Datadog RUM 과금 구조와 세션 개념에 대해 이해 필요

- Session은 사용자 별로 생성됩니다. (접속자 많을 수록 비용이 올라가는 구조)
- Session은 End User가 아무런 동작이 없으면 15분 후 소멸
- Session은 End User가 지속적으로 요청을 만들 경우 최대 4시간까지 유효

2. Datadog의 권장 Tag를 활용하세요

- service, env, version tag 기준으로 분석 데이터를 뽑는 경우가 많습니다

3. RUM event를 선별적으로 수집할 수 있도록 RUM 초기화 블록에 조건문을 추가

- 특정 등급의 End User일 경우만 수집되도록 조건문으로 추가하여 선별적 수집
- sampleRate도 동적으로 제어할 수 있도록 구현하시면 예산에 맞춰 RUM을 사용하시는데 도움이 됩니다.

4. RUM event와 Backend 요청을 연결하여 볼 수 있도록 옵션 설정

- allowedTracingOrigin 옵션에 연결하고자 하는 도메인을 넣으면 APM Trace와 연계하여 분석이 가능합니다.

```
import { datadogRum } from '@datadog/browser-rum';

datadogRum.init({
    applicationId: '<DATADOG_APPLICATION_ID>',
    clientToken: '<DATADOG_CLIENT_TOKEN>',
    site: '<DATADOG_SITE>',
    service: 'my-web-application',
    env: 'production',
    version: '1.0.0',
    sampleRate: 100,
    replaySampleRate: 100,
    trackInteractions: true
    allowedTracingOrigins: ["https://api.example.com",
    /https://V.*\my-api-domain\.com/]
});

datadogRum.startSessionReplayRecording();
```

#Quick Review

Q1. 서비스 사용자 관점의 모니터링이 필요할때 고민해 볼 수 있는 Datadog Product는? (2개)

1. APM
2. NPM
3. RUM
4. Synthetics

RUM, Synthetics

Q2. Browser RUM event와 APM의 Trace를 연결하는데 사용되는 옵션은??

1. allowedTracingOrigins
2. trackInteractions
3. sampleRate
4. service

```
import { datadogRum } from '@datadog/browser-rum'

datadogRum.init({
  applicationId: '<DATADOG_APPLICATION_ID>',
  clientToken: '<DATADOG_CLIENT_TOKEN>',
  ...otherConfig,
  allowedTracingOrigins: ["https://api.example.com", /https:\/\/\.*\.my-
  api-domain\.com/]
})
```



Datadog을 학습하기 위한 플랫폼



Courses

Home / Courses

Running Into An Issue?

If you are experiencing any issues with the platform, there are two ways to reach out to the Learning team at Datadog. First, there is our email address learn@datadoghq.com. Then there is the #training slack channel. If you don't have an account there, go to <http://chat.datadoghq.com> and sign up.

A screenshot of the "Courses" page. At the top is a search bar with "Search courses" and a "Go" button. Below it is a list of 15 courses, each with a small circular icon to its right. The courses are: "Introduction to Datadog", "Understanding Observability", "Tagging Best Practices", "Introduction to Monitoring", "Building Better Dashboards", "Introduction to Application Performance Monitoring", "Introduction to Log Management", "Going Deeper with Logs: Processing", "Introduction to Integrations", "Introduction to Service Level Objectives", and "Introduction to Monitoring Kubernetes".

▼ Courses (15)

- Introduction to Datadog
- Understanding Observability
- Tagging Best Practices
- Introduction to Monitoring
- Building Better Dashboards
- Introduction to Application Performance Monitoring
- Introduction to Log Management
- Going Deeper with Logs: Processing
- Introduction to Integrations
- Introduction to Service Level Objectives
- Introduction to Monitoring Kubernetes

▼ Collapse all

❖ Datadog 핸즈온 램

➤ <https://learn.datadoghq.com>

❖ 24시간 서포트 지원 센터

➤ support@datadoghq.com

실제 사이트에서 RUM 맛보기

우리 서비스에 RUM을 적용했을 때 볼 수 있는 데이터?

shigeya-dd/rum-chrome-extension (Github 프로젝트)

- <https://github.com/shigeya-dd/rum-chrome-extension>

What is it?

- 실서비스에 RUM SDK 적용없이 Datadog RUM 기능을 확인해볼 수 있는 Chrome-extension

주요 설정

- Contents.js 에 Datadog RUM 인증정보 저장
- Manifest.json 트래킹하고자 하는 도메인 입력
- Chrome에서 Extension 로드

기대하는 결과

- 해당 사이트 접속시 페이지 로딩 성능 및 Action 트래킹되어 Datadog Portal에서 확인이 가능합니다

Chrome extension for Datadog RUM demo

I'm going to keep the previous menuless extension in another repository [here](#) as the minimalist version.

What is this?

- Chrome extension for single function to inject JavaScript snippet for Datadog RUM.
- You can demo/trial Datadog RUM with any live web sites.
- Please check the [document](#) for Datadog RUM.

How to setup

- Clone or download this repository to your PC.
- `contents.js`
 - `applicationId` : Your application ID
 - `clientToken` : Your client token
 - `site` : Datadog site to send RUM
 - `service` : service tag
 - `env` : env tag
 - `setUser()` : Sample code for adding custom attributes
- `manifest.json`
 - `matches` : Site URL for RUM demo
 - Refer [doc](#) for match patterns.
 - `js` : replace `datadog-rum-v3.js` with `datadog-rum-recorder.js` for RUM session replay.
- Download Datadog RUM javascript file
 - You can use `rum_download.sh` file.
- Load extension to your Chrome
 - Open extention page. `chrome://extensions`
 - Turn on *Developer mode* switch.
 - Click *Load unpacked* button and load this folder.
 - *RUM Demo* will appear in the extensions page and the extensions menu.

Q&A