

Datadog Live Seoul 2024

Datadog 2024 Highlight



정영석
기술팀 리더



조용원
Sales Engineer



이누리
Sales Engineer

2024 Datadog Live Highlight

Agenda

1 2024 한국 고객분들의 사용 트렌드 Top3

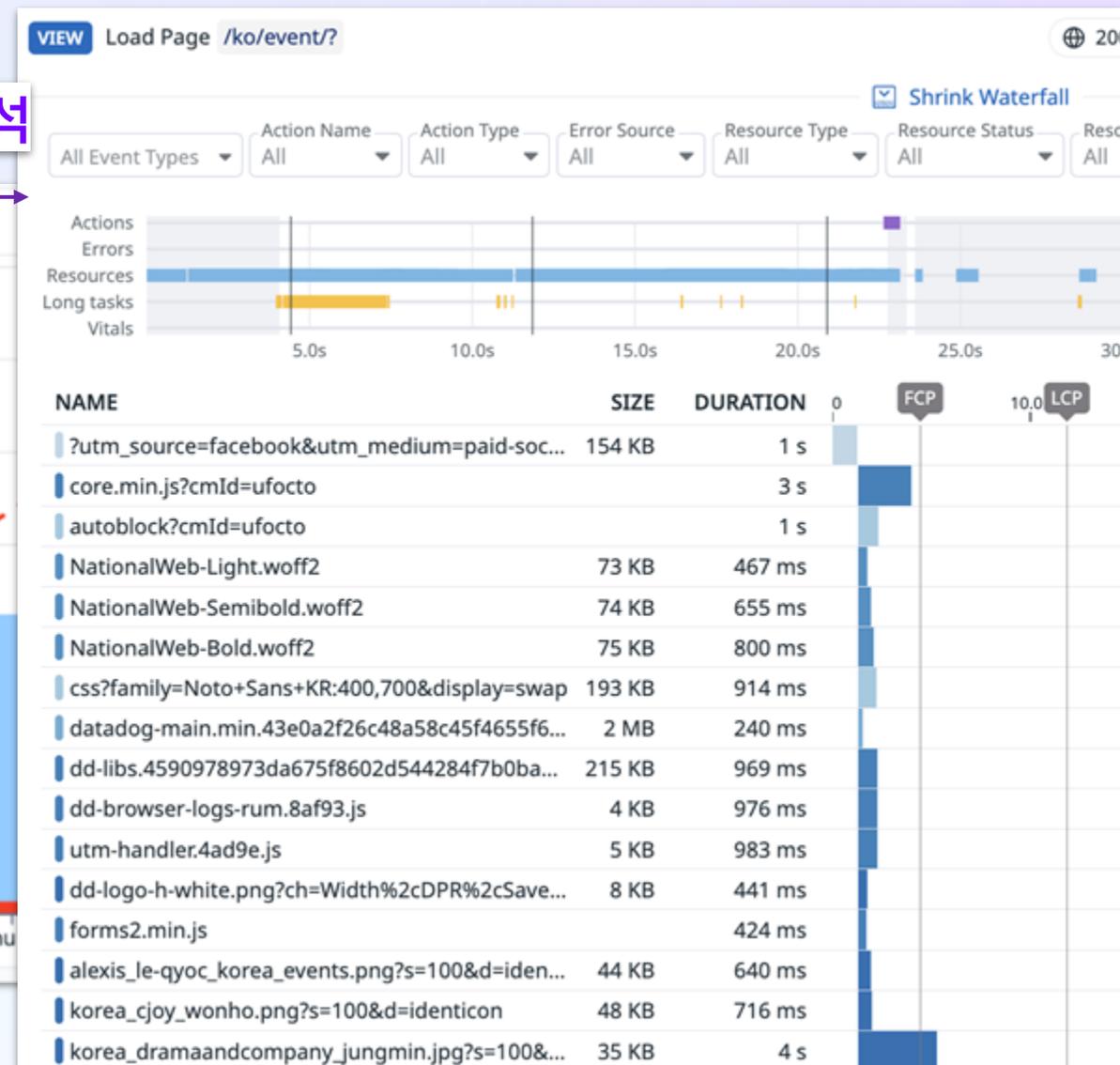
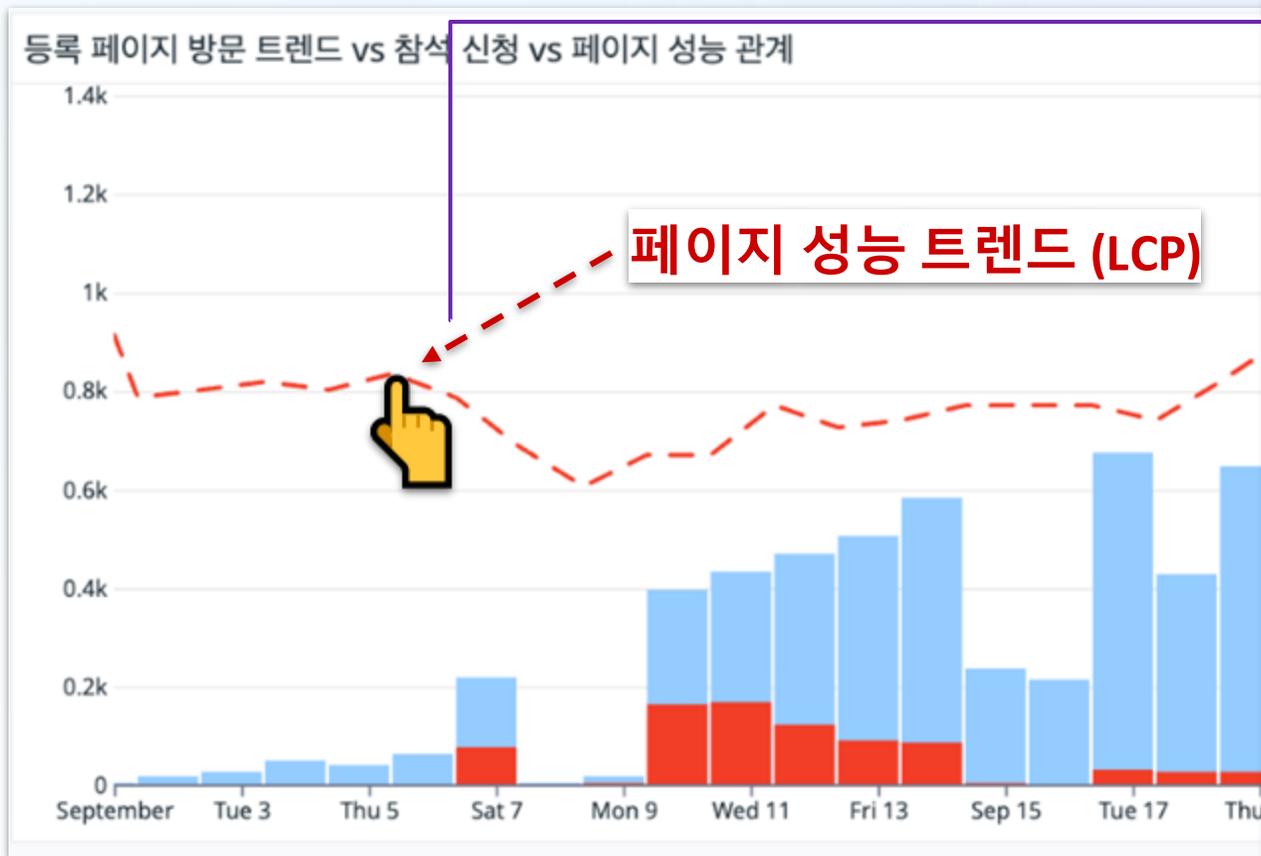
2 직무별 (개발자, SRE, AI) 주요 신기능

2024 한국 고객분들의 사용 트렌드 ★ Top 3

#1 Datadog을 이용한 사용자 모니터링 증가 (End-to-End)

1) UX 성능 및 여정 분석이 쉬워졌어요

성능이 느려진 시점에 대한 Deep Dive 분석



#1 Datadog을 이용한 사용자 모니터링 증가 (End-to-End)

2) 고객 voc 분석이 수월해졌어요

실제 고객 화면 재현 분석

The screenshot displays the Datadog browser session replay interface. The main window shows a browser session with the URL `/booking/registerPassenger`. The page content is in Korean, showing a booking form with fields for name, date of birth, and phone number. A mouse cursor is positioned over a '다음' (Next) button. The right-hand panel provides session details and a list of events. The events list includes:

- 0 s: Load Page /booking/registerPassenger
- 8.02 s: click on 다음 on page /booking/registerPassenger
- 10.6 s: Load Page /booking/extras
- 18.7 s: User Focus On /booking/extras
- 18.8 s: click on 32열C (JINI FAST) on page /booking/extras
- 21.6 s: click on 선택완료 on page /booking/extras
- 26.8 s: click on 6열C (JINI PLUS Seat) on page /booking/e...
- 28.4 s: click on 선택완료 on page /booking/extras
- 35.5 s: click on 선택완료 on page /booking/extras

출처: 진에어의 프론트 모니터링 웨비나 발췌

#1 Datadog을 이용한 사용자 모니터링 증가

잠시 후 세션에서 실제 고객 사례로 상세한 내용을 확인해보세요!

트랙1 Observability Mastery (고도화 사례)	트랙2 Observability Synergy (시너지 활용 사례)	트랙3 Observability Journey (도입 여정 사례)
CJ올리브영 통합된 Observability를 통한 점진적 비즈니스의 개선	SK C&C SK그룹의 모니터링 환경 표준화와 Cloud 운영 서비스 최적화 여정	프롬더레드 한 달로 완성하는 End-to-End 모니터링과 데이터 인사이트로 플랫폼 고도화
드라마앤컴퍼니 12년차 스타트업의 아키텍처 리팩토링 돌아보기	LG유플러스 LG유플러스 Enterprise 환경에서의 Observability 거버넌스	뉴넥스 3개월 차 DevOps팀의 Datadog 활용
휴식 시간		
삼성전자 데이터 안전 수송 작전: From Samsung To Datadog	체카 Datadog과 함께한 서비스 위기 탈출과 클라우드 인프라 운영	아모레퍼시픽 뷰티 이커머스 테크 기업이 시스템 고도화/안정화를 위한 Front-end Observability 적용 사례
딤브레인 Datadog으로 사용자 경험 완성하기	Datadog LLM Observability를 확보하기 위한 Datadog 활용 방법	리코 웨이스트 매니지먼트 스타트업 프론트엔드 개발자의 Datadog을 활용한 효율적인 모니터링 시스템 구축기

#2 비즈니스 KPI와 IT 성능 지표 연계 분석

CS팀

결제가 안된다고 계속 새로운 케이스가 올라오고 있어요!

경영팀

1시간 동안 매출에 영향이 있네요!



운영팀

운영 대시보드에서는 특이 사항이 없어 보여요.

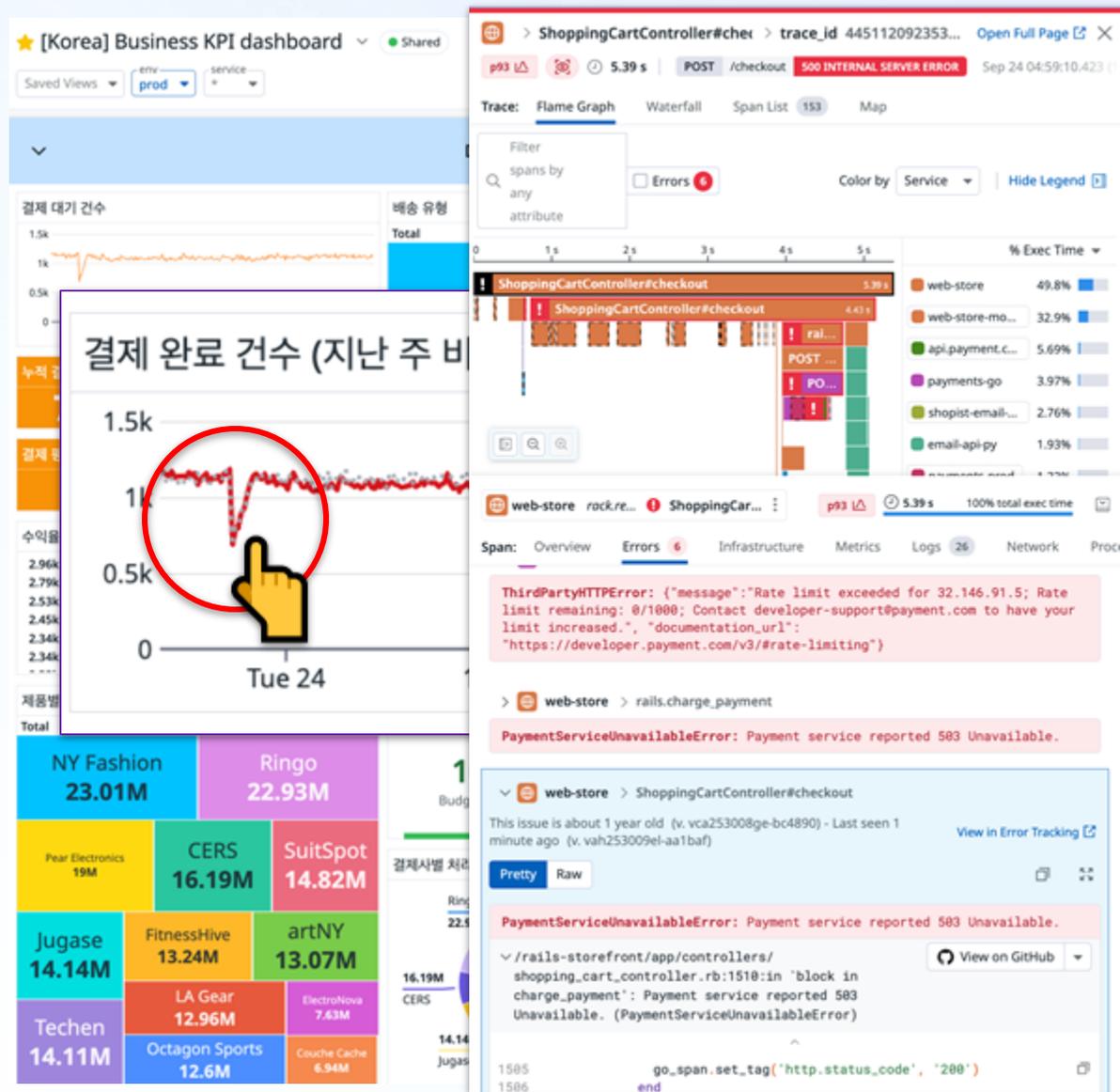
매출이 언제부터 내려갔나요?

개발팀

제 자리에서는 잘 됩니다.

에러 지표도 괜찮아 보입니다. 프론트/백엔드 정상입니다.

#2 비즈니스 KPI와 IT 성능 지표 연계 분석



함께 분석하면 무엇이 좋나요?

- 이슈에 따른 비즈니스 임팩트 분석
- 성능 개선에 따른 비즈니스 효과 분석

📌 비즈니스와 엔지니어링팀의 협력 증가

📌 시스템의 빠른 피드백 및 반영

📌 비즈니스 마인드셋을 가진 엔지니어링 팀 구

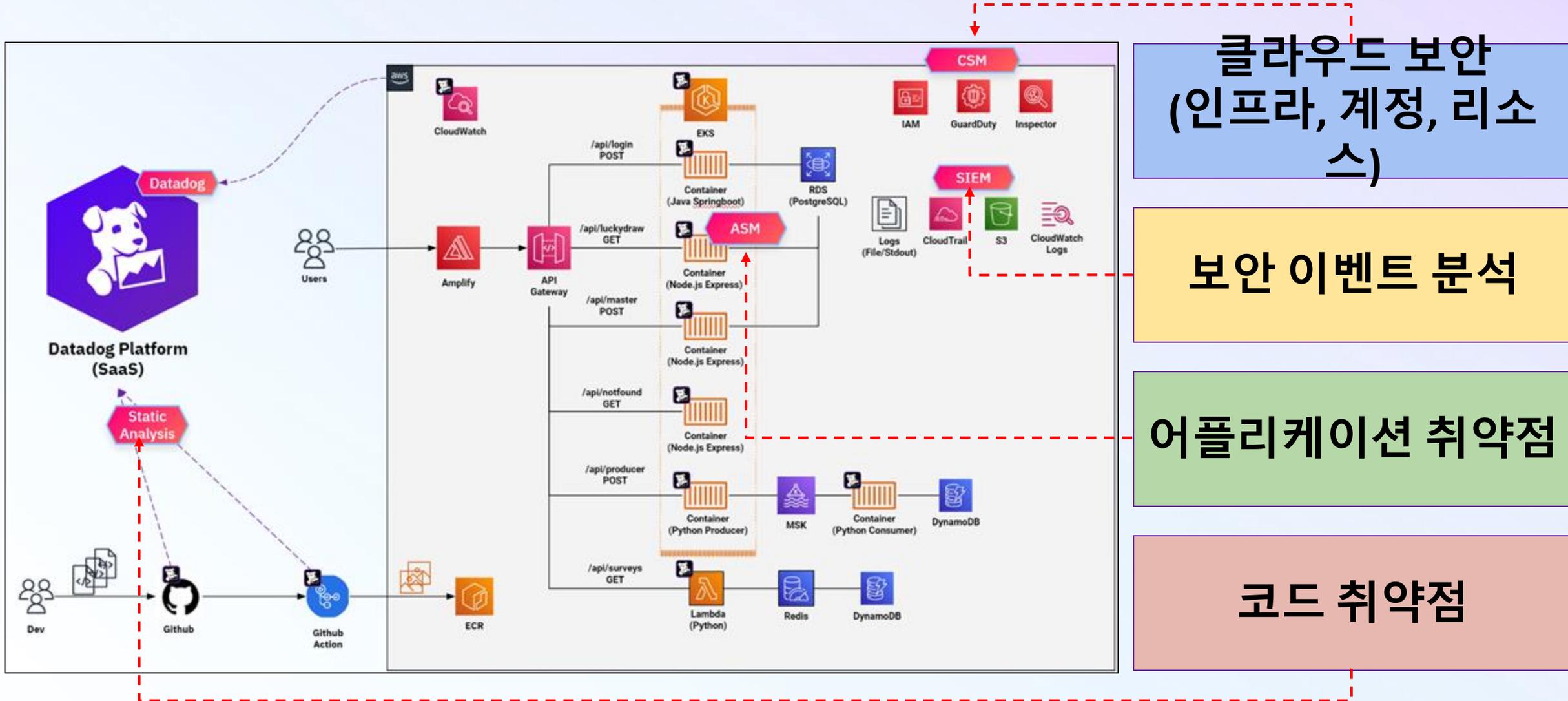
성

#2 비즈니스 지표 연계 분석

잠시 후 세션에서 실제 고객 사례로 상세한 내용을 확인해보세요!

트랙1 Observability Mastery (고도화 사례)	트랙2 Observability Synergy (시너지 활용 사례)	트랙3 Observability Journey (도입 여정 사례)
CJ올리브영 통합된 Observability를 통한 점진적 비즈니스의 개선	SK C&C SK그룹의 모니터링 환경 표준화와 Cloud 운영 서비스 최적화 여정	프롬더레드 한 달로 완성하는 End-to-End 모니터링과 데이터 인사이트로 플랫폼 고도화
드라마앤컴퍼니 12년차 스타트업의 아키텍처 리팩토링 돌아보기	LG유플러스 LG유플러스 Enterprise 환경에서의 Observability 거버넌스	뉴넥스 3개월 차 DevOps팀의 Datadog 활용
휴식 시간		
삼성전자 데이터 안전 수송 작전: From Samsung To Datadog	체카 Datadog과 함께한 서비스 위기 탈출과 클라우드 인프라 운영	아모레퍼시픽 뷰티 이커머스 테크기업이 시스템 고도화/안정화를 위한 Front-end Observability 적용 사례
딥브레인 Datadog으로 사용자 경험 완성하기	Datadog LLM Observability를 확보하기 위한 Datadog 활용 방법	리코 웨이스트 매니지먼트 스타트업 프론트엔드 개발자의 Datadog을 활용한 효율적인 모니터링 시스템 구축기

#3 보안팀도 함께 사용하는 Datadog



클라우드 보안
(인프라, 계정, 리소스)

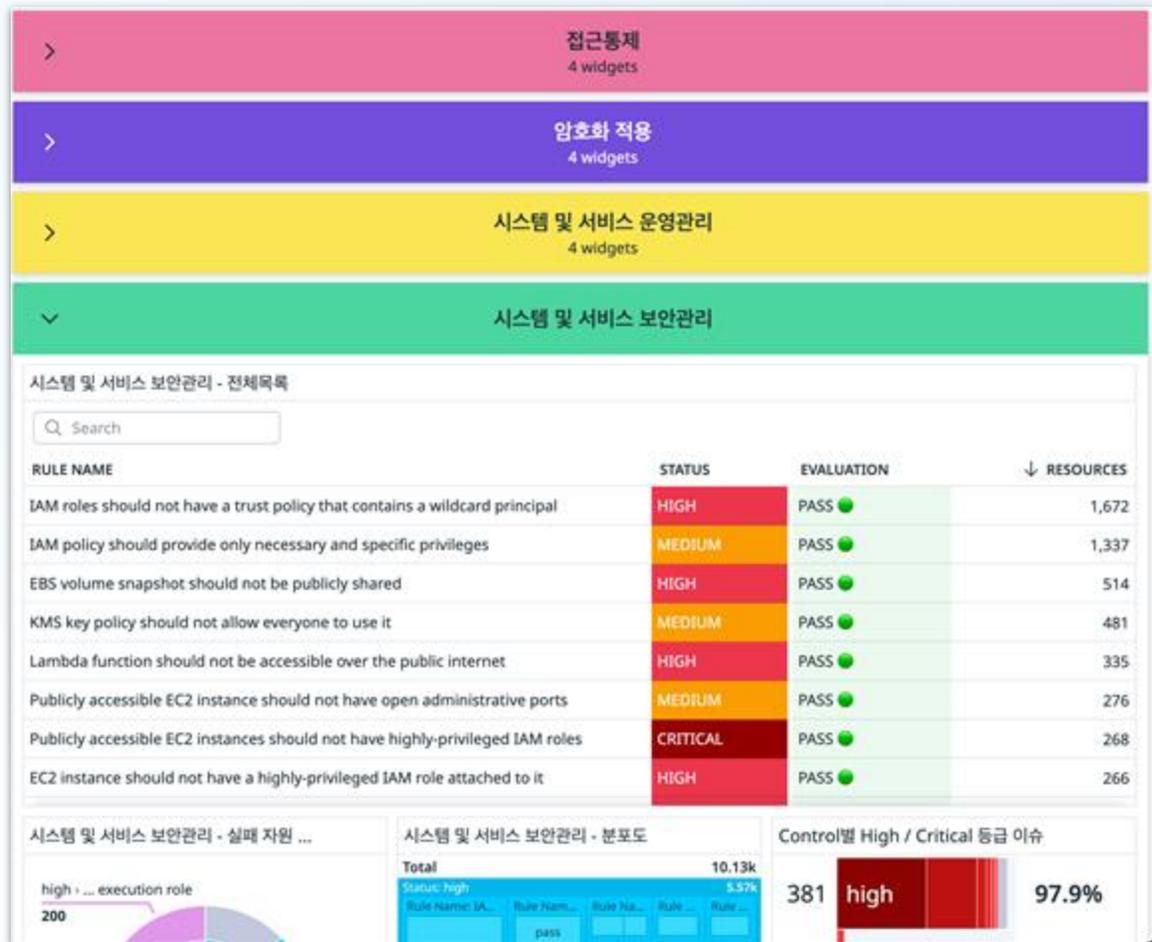
보안 이벤트 분석

어플리케이션 취약점

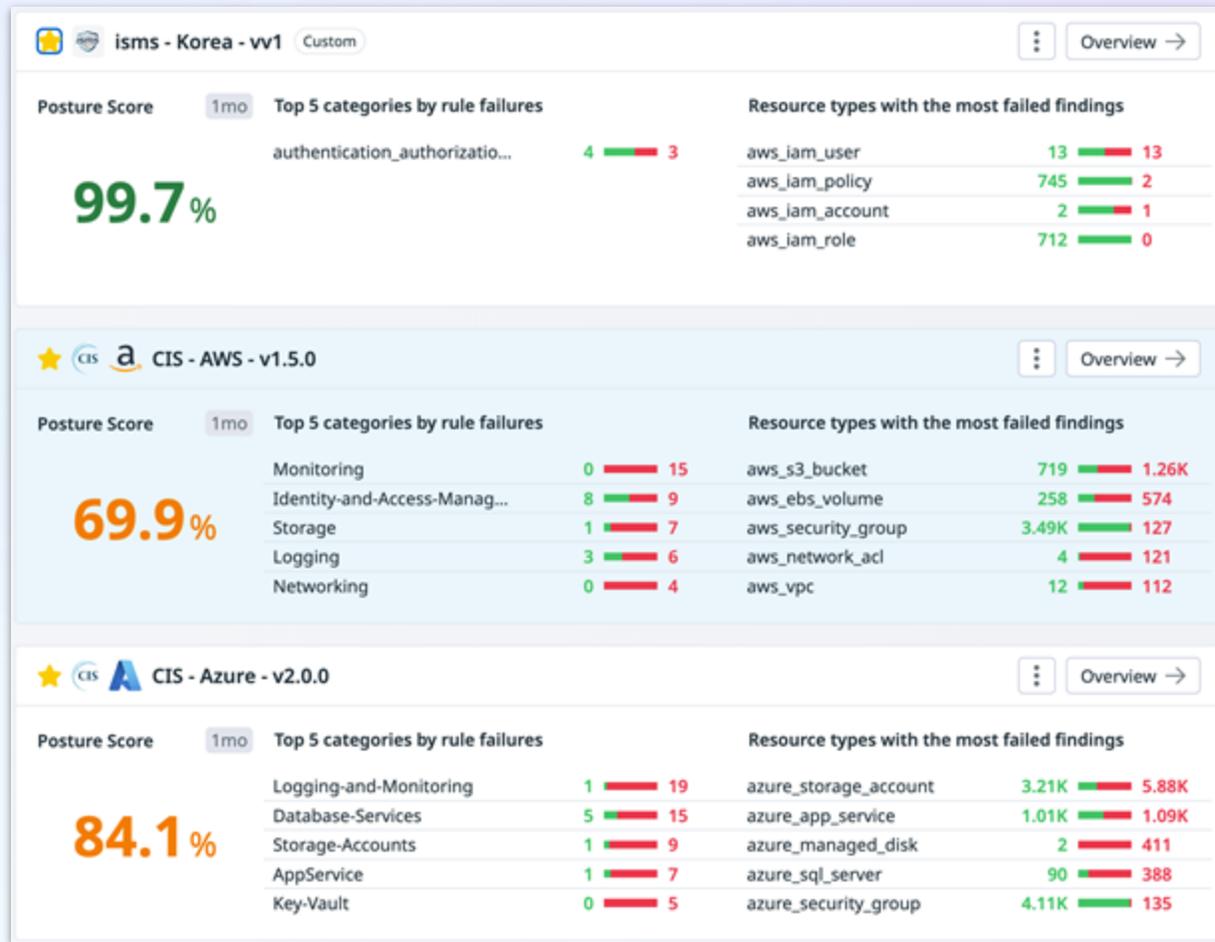
코드 취약점

#3 보안팀도 함께 사용하는 Datadog

보안 감사 항목별 점검 대시보드

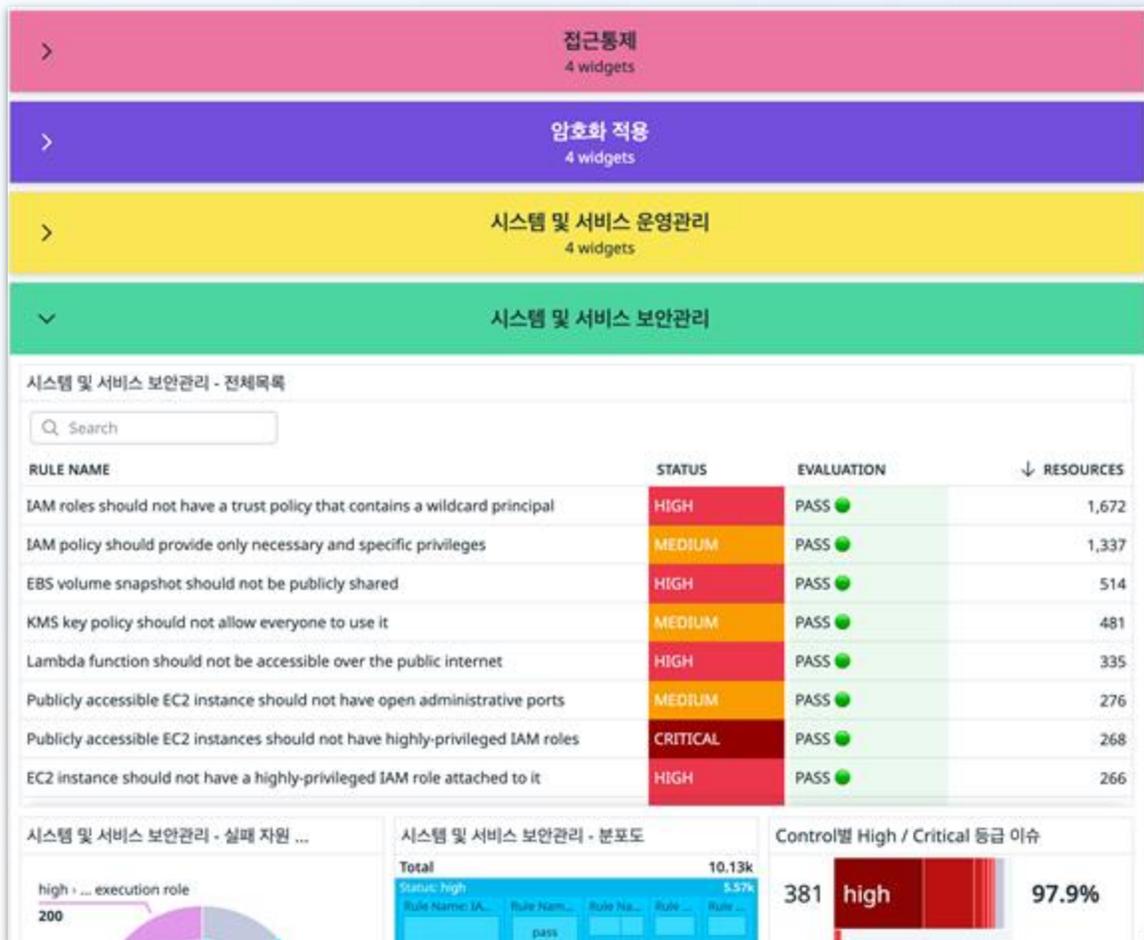


다양한 보안 프레임워크에 대한 준수 여부 확인



#3 보안팀도 함께 사용하는 Datadog

보안 감사 항목별 점검 대시보드



보안과 Observability가 통합되면 무엇이 좋나요?

- 보안 취약점에 따른 서비스 임팩트 분석 용이
- 보안팀과 개발팀의 사일로 개선



성공적인 DevSecOps의 첫걸음



Datadog 부스 투어



소개

- Datadog 3분 요약정리
- Datadog 핵심 포인트



보안

- 보안 기능이 뭐가 있나요?
- Datadog 보안, 왜 좋나요?



Observability



End-to-End

- 프론트부터 인프라까지
- 장애 시간의 최소화

생성형 AI



LLM 모니터링

- GenAI, 어떻게 모니터링할까?
- 에러? 성능? 품질? 보안?



Security

한국 고객분들이 눈여겨 볼만한 7가지 주요 기능

💡 개발자들의 삶을 도와주는 New Feature 3가지

★ Live Debugger

✅ 에러 트랜잭션에서 이용된 요청/응답 파라미터를 바로 확인해보

The screenshot displays the Visual Studio Live Debugger interface with three main panels:

- Left Panel (C# Code):** Shows the `FinalCheck` method in `CheckoutController.cs`. A red dot indicates a `System.Exception: Price can't be less` error at line 111. A yellow box highlights the `item` variable in the `foreach` loop.
- Middle Panel (C# Code):** Shows the `CheckoutController.Index()` method. A yellow box labeled "결제 서비스" (Payment Service) highlights the `httpClient.PostAsync` call. A data view shows the `updatedShopCartItems` list with one item: `CartItemViewModel` (CatalogItemId: 1, ProductName: Laptop, Price: -0.010, Quantity: 1).
- Right Panel (Python Code):** Shows the `apply_coupon()` function in `views.py`. A yellow box labeled "쿠폰 서비스" (Coupon Service) highlights the `discount` calculation logic.

At the bottom, the **Values at** window is open, showing the state of variables at the point of the exception:

- `this = CheckoutController`
- `model = CheckoutViewModel`
- `@exception = Exception`
- `responseBody = {"final_price": -0.0099999999999999787, "adjusted_items": [{"id": "1", "name..."}`
- `response = HttpResponseMessage`
- `applyCouponResponse = ApplyCouponResponse`
- `updatedShopCartItems = List<CartItemViewModel> (size: 1)`
- `serializedJson = String`
- `content = StringContent`

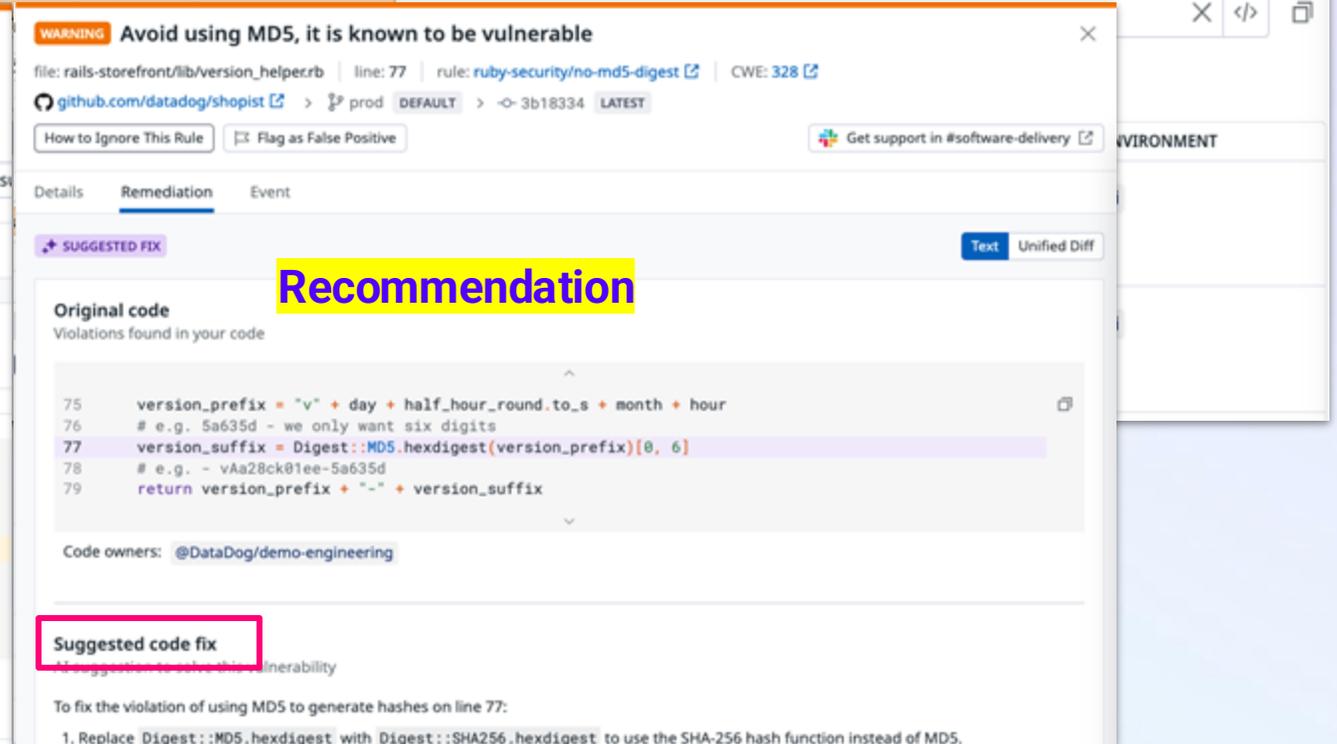
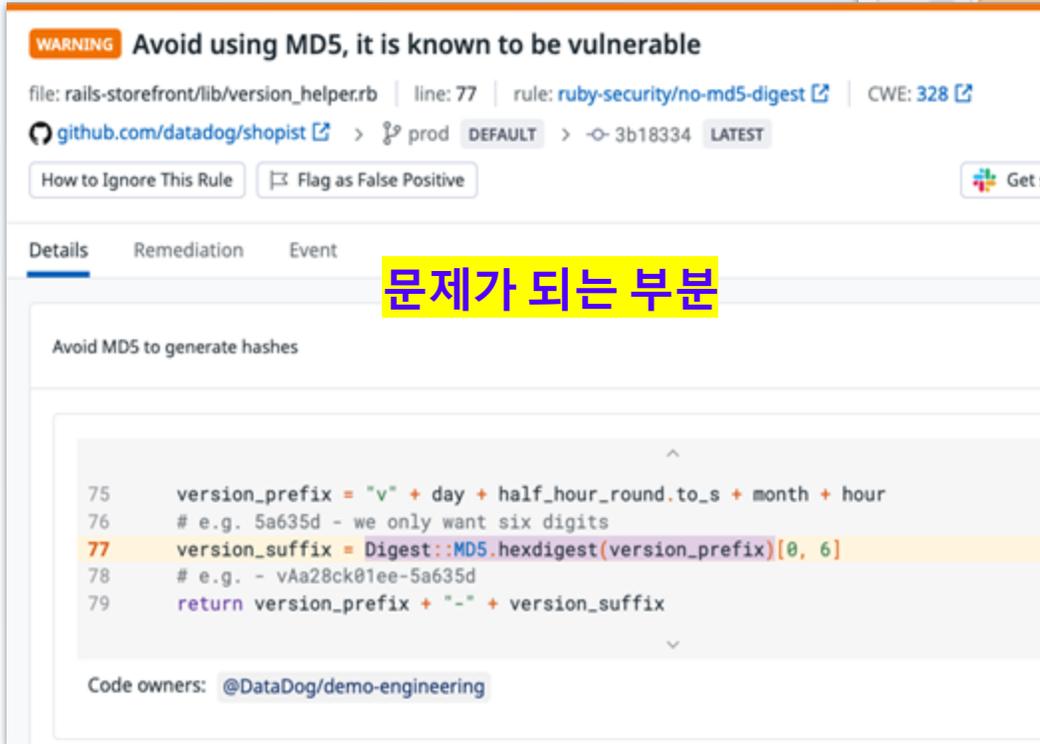
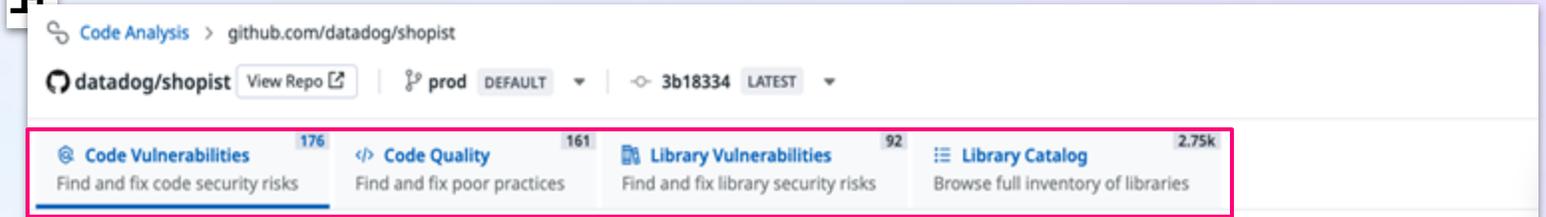
On the right side of the **Values at** window, the state of the request and response objects is shown:

- `request = WSGIRequest`
- `cursor = Psycopg2TracedCursor`
- `conn = Psycopg2TracedConnection`
- `item = dict (size: 4)`
- `coupon_code = '10DollarOFF'`
- `item_units = 1`
- `data = dict (size: 2)`
- `coupon = Record`
- `item_price = 9.99`
- `discount = 10.0`

💡 개발자들의 삶을 도와주는 New Feature 3가지

- ★ Live Debugger
- ★ Mobile App Testing
- ★ **Code Analysis**

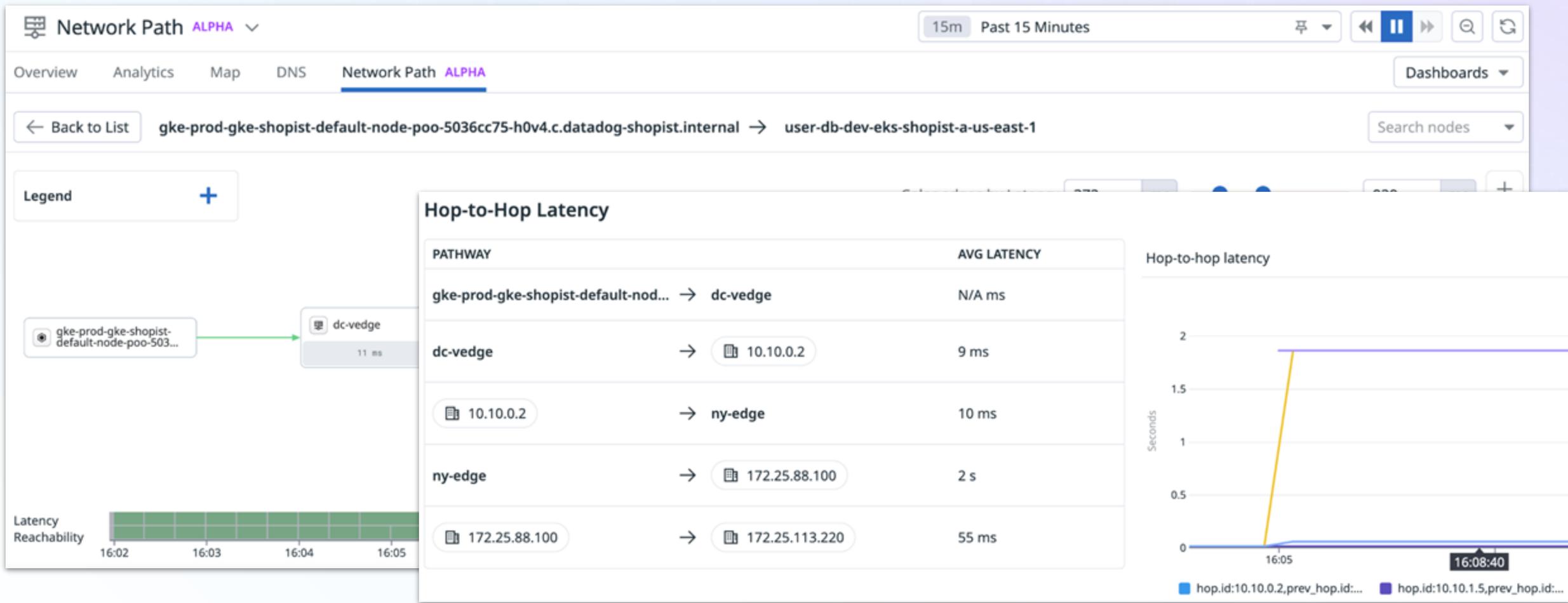
✅ Repository 연동만으로 코드 정적분석 및 취약성 분석을 손쉽게 해보세요



💡 SRE/인프라팀의 삶을 도와주는 New Feature 2가지

★ Network Path

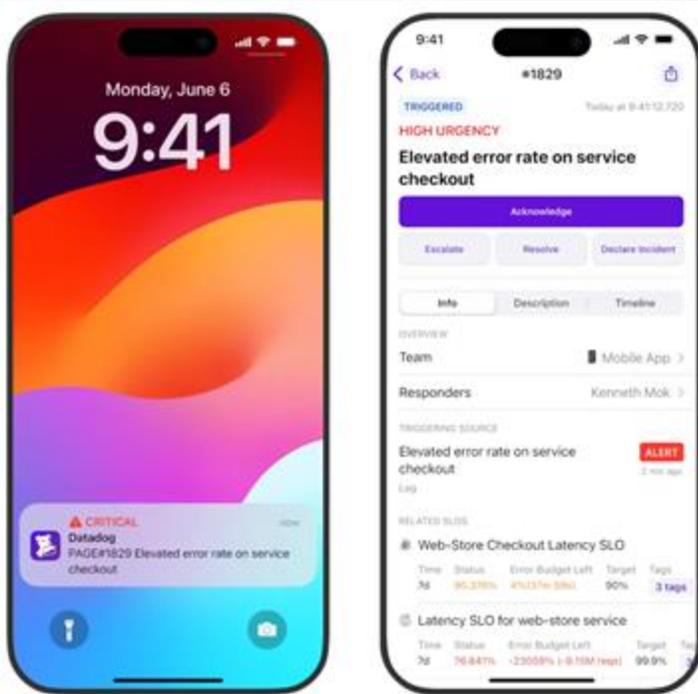
✅ Network 경로의 Hop간 레이턴시를 분석할 수 있어요



💡 SRE/인프라팀의 삶을 도와주는 New Feature 2가지

★ Network Path

★ On-Call



✅ 긴급상황 발생 시 담당자에게 조건에 맞게 알려줘요 📞

On-Call Summary Teams Pages Schedules Escalation Policies Analytics My On-Call Profile

Search...

My Schedules Hide Controls Showing 1-6 of 6 Schedules + New Schedule

All Schedules Today < > Jun 25, 2024 - Jul 1, 2024 View in America/New York (UTC-05:00)

My Teams

Search facets

Service

- fse-auto-process 1
- redis-cache 1
- auth-resolver 1
- checkoutservice 1
- payment-gateway 1

Team

- mobile-app 3
- cart 2
- front-end 1

Primary On Call

Mobile App Currently On Call: Alan Fineberg Escalation Policies: Default Escalation Policy

Tuesday 25 Wednesday 26 Thursday 27 Friday 28 Saturday 29 Sunday 30 Monday 1

Alan Fineberg Nicolas Mulet

Secondary On Call

Mobile App Currently On Call: Quentin Litzler Escalation Policies: Default Escalation Policy

Tuesday 25 Wednesday 26 Thursday 27 Friday 28 Saturday 29 Sunday 30 Monday 1

Quentin Litzler Yassir Ramdani

Business Hours

Mobile App Currently On Call: Guille González Escalation Policies: Default Escalation Policy

Tuesday 25 Wednesday 26 Thursday 27 Friday 28 Saturday 29 Sunday 30 Monday 1

Guille G Kenneth Guille G Kenneth Guille G Kenneth Guille G Kenneth Chirag K Brandon

Business Hours

Front End Currently On Call: No one is on-call Escalation Policies: Default Escalation Policy

Tuesday 25 Wednesday 26 Thursday 27 Friday 28 Saturday 29 Sunday 30 Monday 1

Samant Samant Samant Samant Daniel N

Business Hours

Cart Currently On Call: Brandon Smith Escalation Policies: Default Escalation Policy

💡 빅데이터/AI 팀의 삶을 도와주는 New Feature 2가지

★ Data Jobs Monitoring

✅ 느린 Spark Job을 식별하고, Stage별 특이사항을 확인 할 수 있어요

The image displays two screenshots from the Databricks interface. The left screenshot shows the 'Data Jobs Monitoring' dashboard, which includes a search bar, alert status (2 ALERT), job run statistics (9.67k Job runs, 7.81% Failed, 4 min 59 s avg duration, \$2,176.14 est costs), and a table of job runs. The right screenshot shows a detailed trace for a Spark job, highlighting a slow stage (Stage 4550069) with a 17 min 40 s total executor time. The trace includes a flame graph, a table of spans, and a detailed view of the slow stage showing its breakdown by category (Compute, IO, Shuffle, GC) and a diagram of the Spark execution plan.

Data Jobs Monitoring Dashboard:

- 2 ALERT Monitors alerting
- 4 jobs with failures
- 9.67k Job runs (7.81% Failed)
- AVG DURATION: 4 min 59 s (75% ⬆)
- EST COSTS: \$2,176.14 (11% ⬇)

TYPE	NAME	MONITORS	LAST STATUS
Batch Jobs	volume-shipping-job	1 OK	OK
Streaming Jobs	spark-pi		OK
Clusters	product-insights	2 ALERT	OK

Spark Job Trace Details:

- pricing_summary_report > pricing_summary_report > trace_id 1722860125064879891
- 3 min 1 s | Sep 24 16:01:18.408 (25m ago)
- Trace: Flame Graph, Waterfall, Span List (30), Map
- Filter spans by any attribute
- Color by: Service
- Number of input batches: 7.33k
- Number of output rows: 30.0M
- Stage 4550069: 17 min 40 s (99.7% of query)
- Stage 4550152: 1.3%

Category	Percentage
Compute	95.9%
IO	2.4%
Shuffle	0.0%
GC	1.7%

💡 빅데이터/AI 팀의 삶을 도와주는 New Feature 2가지

★ Data Jobs Monitoring

★ LLM Observability

✅ LLM 요청의 성능, 에러 원인과 답변의 퀄리티를 분석할 수 있어요 🤖

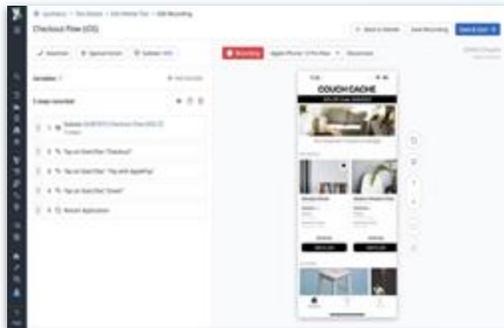
The image displays the Datadog LLM Observability interface. On the left, the 'LLM Monitoring' dashboard shows a 'Catalog' with filters for Model Provider (OpenAI, Anthropic, Cohere, PaLM), Env (datadog-dash-demo, dev, prod, staging), and Service (chatbot, concurrent-correlator, oai-server). A 'Total Requests' chart shows a peak on Thursday. The main area shows a list of applications with a search filter for 'user_handle:lively.hase@dog.com'. A detailed trace for 'shopist-chat-v2' is shown on the right, highlighting a 'plan' step with a 'plan' error. The trace details include input/output, generated topics, and a stack trace for 'openai.APISConnectionError: Connection error.' At the bottom, a summary bar shows '1 ALERT', '0' requests, '1.4 req/s', '1.03 s', and '6.0 tok/rec'.

💡 직무별 주요 신기능 정리



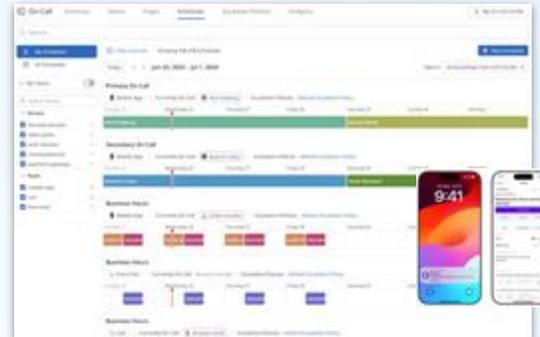
개발자

- 1 Live Debugger [Link](#)
- 2 Mobile App Testing [Link](#)
- 3 Code Analysis [Link](#)



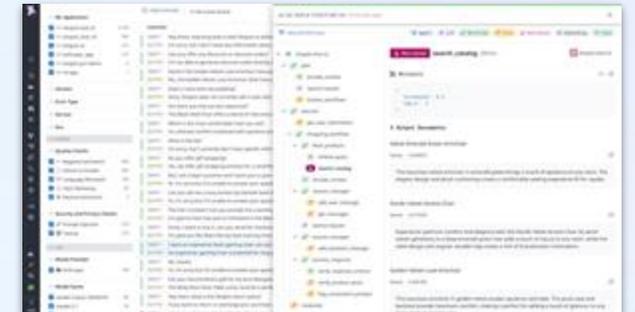
SRE / 인프라

- 1 Network Path [Link](#)
- 2 On-Call [Link](#)



빅데이터 / AI

- 1 Data Jobs Monitoring [Link](#)
- 2 LLM Observability [Link](#)



감사합니다



DATADOG