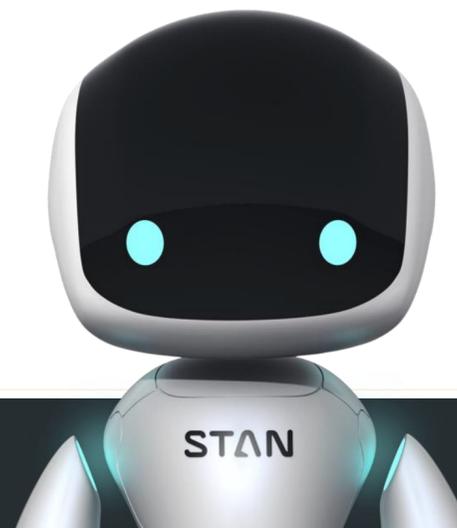


# INSTANA

99% 자동화를 통해 DevOps 환경의 Microservice에 최적화된,

어플리케이션 성능분석 솔루션 (APM)



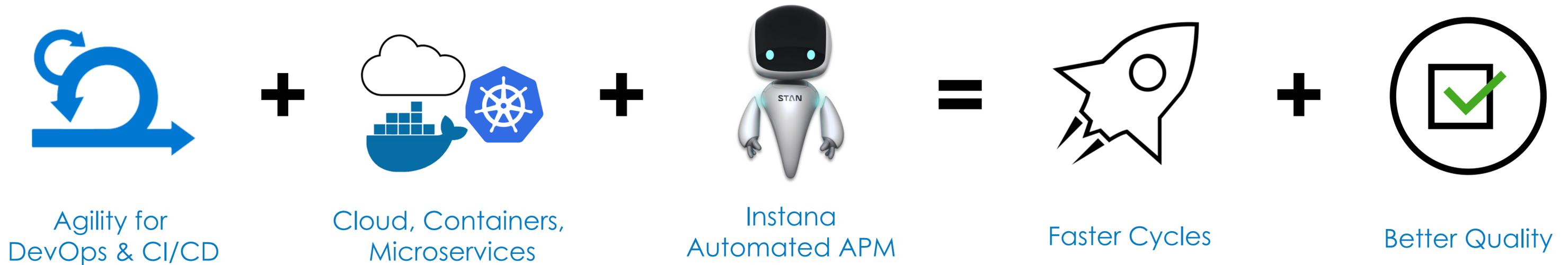
- 2015년 설립, DevOps 환경에 **최적화된** APM 서비스 제공
  - Silicon Valley(본사), 독일(Dev & Support Center), 미국, 독일, 영국, 프랑스, 일본(Sale Office)
- 고객사
  - 25+국가, 300+고객사를 보유하고 있으며 매년 지속적으로 고객사 증가
  - 제조, e-Commerce, 유통, 통신, 금융을 포함한 대부분의 산업에서 상용 서비스중

: 주요 고객사들

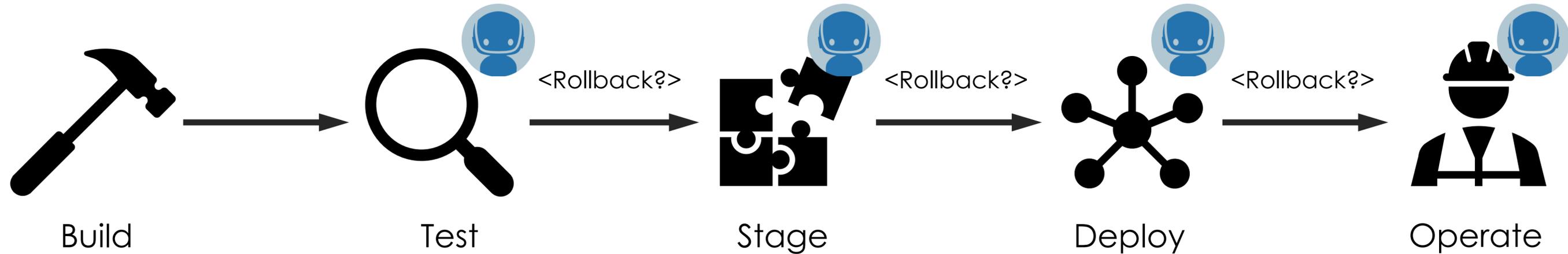


- DevOps에 최적화된 자동 APM으로, APM 구축/운영에 있어 **Manual** 작업을 최소화하고, 사용중인 서비스 Application에 대해 즉각적으로 **직관적인 정보를 제공**함으로써 (Auto Discovery), 장애에 빠르게 대응할 수 있도록 Service Flow에 대한 Insight 및 Agile Feedback 제공

Instana gives them immediate, exact understanding  
with **zero effort!**



- 운영 단계의 Monitoring Tool 포함, Test 단계 → Staging 단계 → 배포 단계까지 Application 변화에 대해 **실시간**으로 직관적인 Feedback 제공



- Test 단계부터 운영(Operate) 영역에 성능과 품질에 대한 즉각적인 Feedback 제공
- 별도의 설정 없이 Full Stack 가시성 제공
- Bug 및 품질 저하의 원인을 직관적으로, 신속히 찾아서 해결할 수 있는 정보 제공

Enabling you to build better software faster!

- 기존 Monitoring 및 APM 솔루션은 구축/운영하는데 상대적으로 많은 Manual 작업이 필요하지만, Instana는 구축/운영에 대부분의 작업이 자동화되어 처리됨

기존 APM



많은 Manual 설치/운영 작업

- Monitoring 대상을 수동으로 선택
- Monitoring을 위해 수동으로 설정
- 수동으로 데이터들에 대한 연관관계를 설정
- 수동으로 alert 규칙과 임계치를 설정
- 수동으로 Issue에 대한 root cause를 분석 결정

VS.

Instana



설정이나 코딩이 필요 없는 자동화된 작업

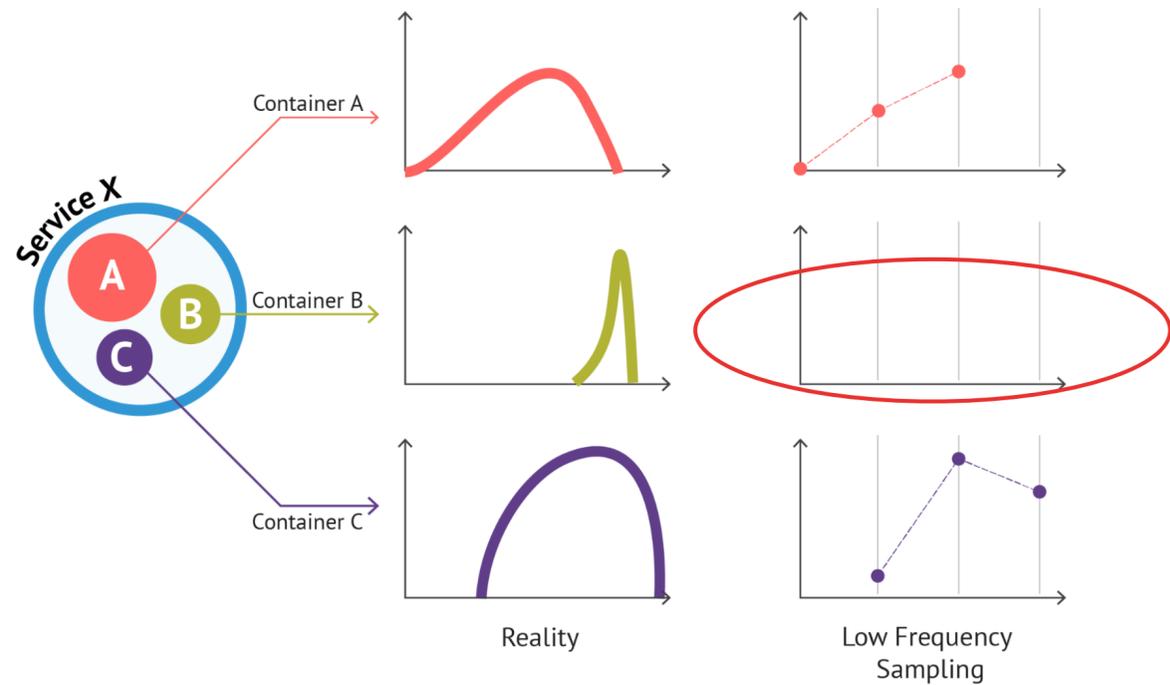
- 모든 대상을 자동으로 찾아서 Monitoring 수행
- 모든 Trace를 자동 추적
- 모든 데이터를 의미있는 정보로 자동 구성
- 새 릴리즈에 대한 즉각적인 가시성 확보 (5초 이내에)
- 성능 Issue 자동 감지 및 alert발생
- 자동으로 Issue에 대한 Root Cause 분석 결정

Traditional monitoring slows down CI/CD cycles and is not able to immediately visualize without significant effort

## 기존 방식(Sampling 방식)

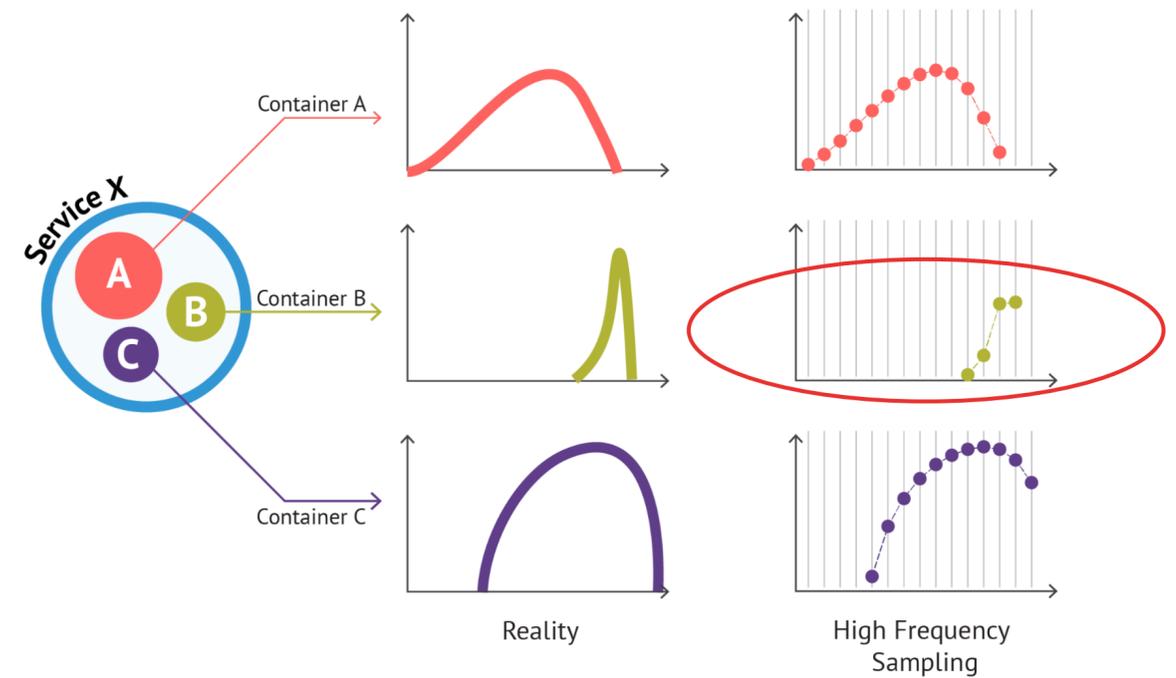
## Container를 위한 최적의 Real Time 방식

### LOW FREQUENCY SAMPLING



visual design by remember to play

### HIGH FREQUENCY SAMPLING



visual design by remember to play

- 기존 Sampling 방식의 데이터 수집 ; every 1-10 minutes
- **Container B Monitoring** 데이터 수집 되지 않음
  - 실시간 Monitoring, Issue Alert 불가

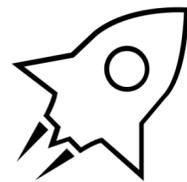
- 데이터 수집 주기를 짧게 하여 빠짐 없이 Monitoring
- Container B 수집 가능
  - 운영 환경 모니터링 불가
  - 방대한 양의 데이터 생산

<https://container-solutions.com/monitoring-performance-microservice-architectures/>

 **edmunds** 미국의 대표적인 Online 자동차 거래 및 분석 정보 서비스 업체 : <https://www.edmunds.com>



Instana 적용한 후, Instana의 즉각적인 Feedback을 바탕으로 **Application의 배포 횟수가 기존 일 1회 → 최대 8회까지** 배포가 가능하게 되고, 이로 인해 발생할 수 있는 오류로 인한 **Alert의 수가 비약적으로 감소**

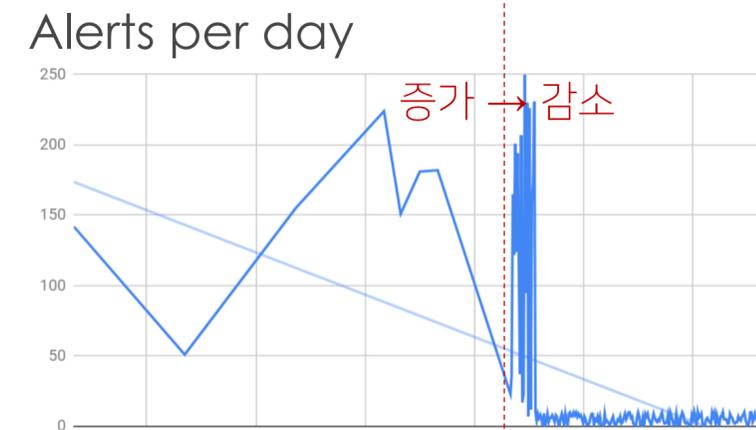
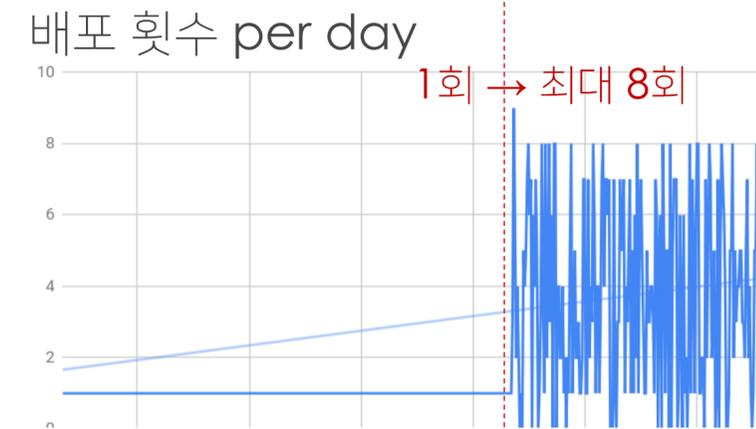


Agile 환경에 맞춰, 지속적이고 빈번한 배포 주기 제공

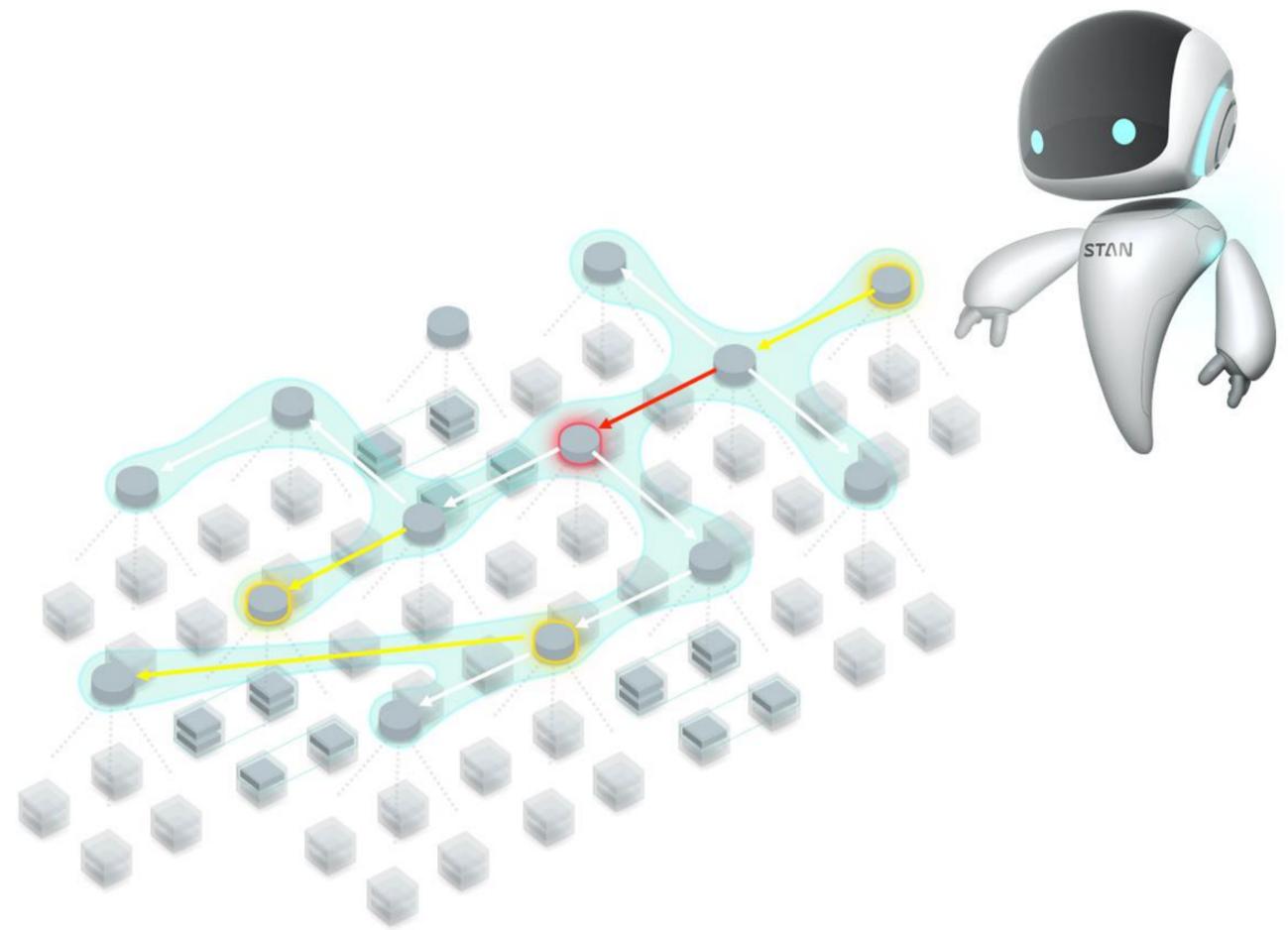


지속적이고 빈번한 배포 환경에 맞게, **서비스 품질 즉시 개선**

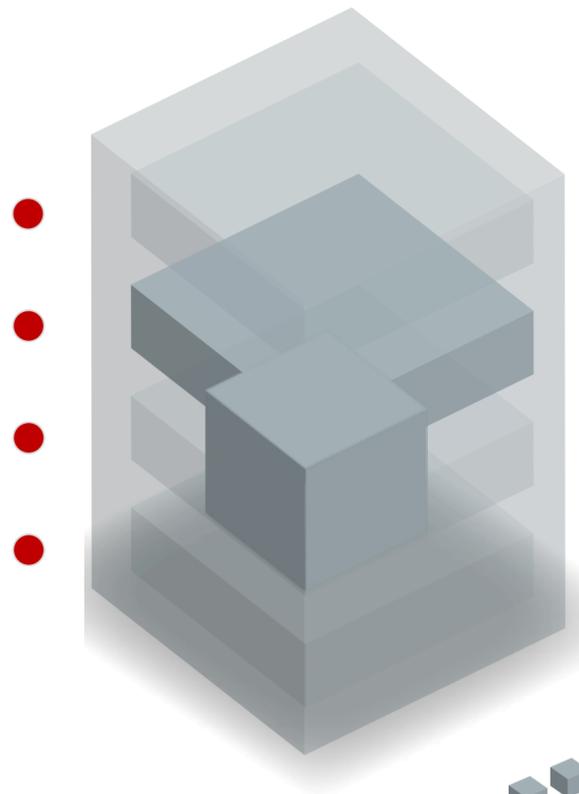
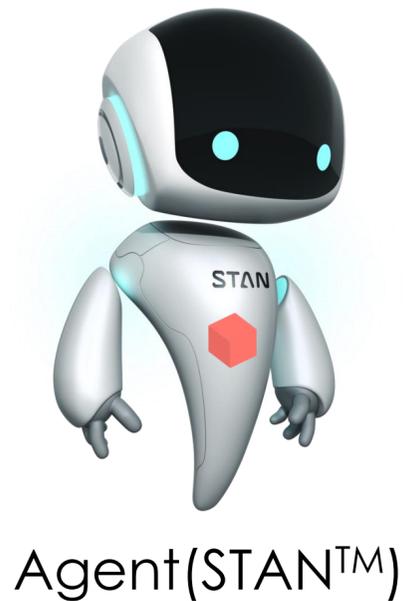
Instana 적용 시점



# Instana 작동 방식

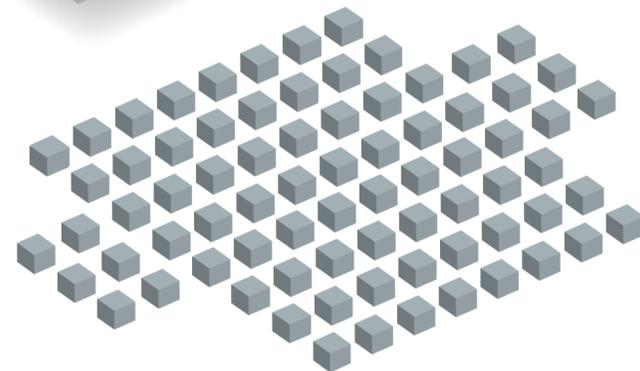


- Host(incl. VM)에 **단 하나의 Agent(STAN™)만 설치**
  - Plug-in 이나 추가 설정없이, Host(VM)내의 모든 기술 Stack을 자동으로 감지하여 Monitoring을 수행하며, Host에 새로운 Component가 추가되더라도 Agent(STAN™)가 자동으로 감지하여 Monitoring 제공



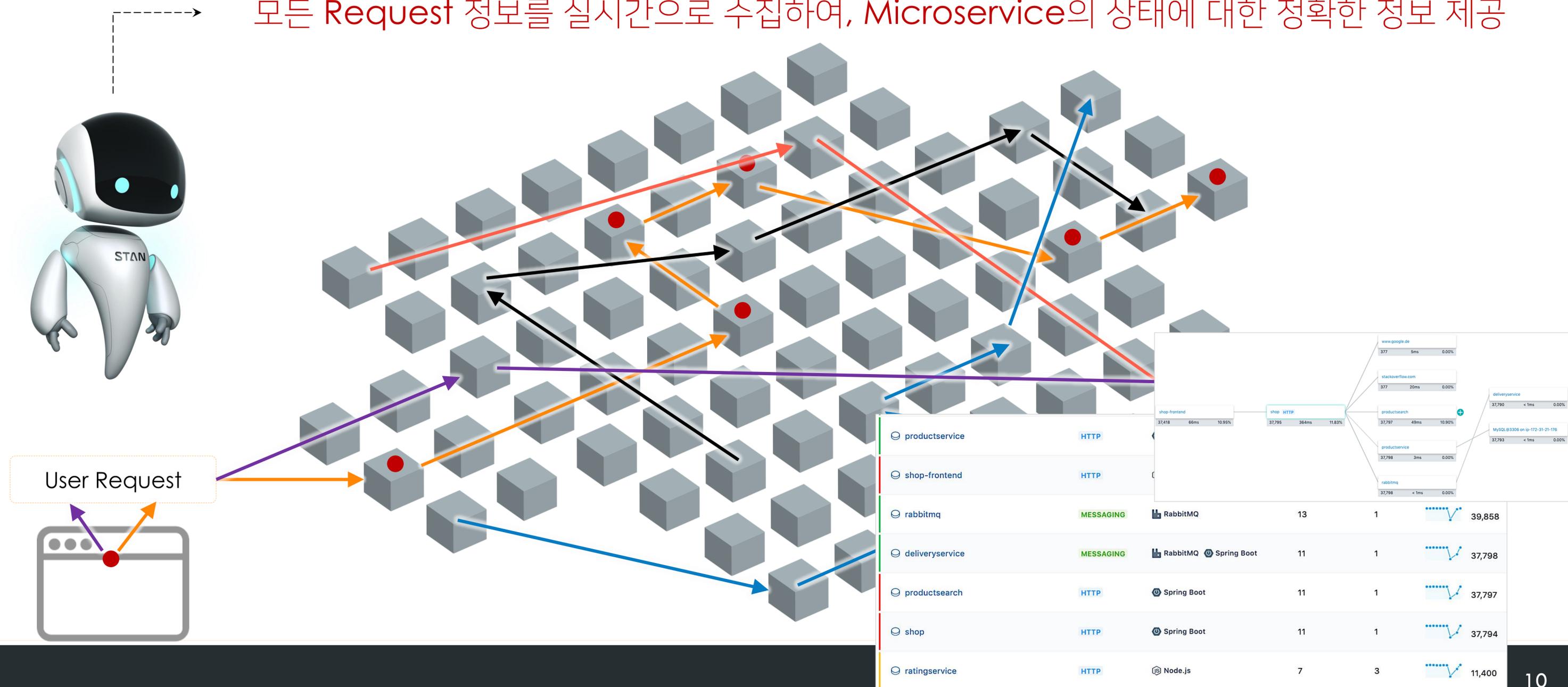
지속적인 실시간 검색 및 Monitoring  
= 즉각적인 가시성 제공

Automatic!  
No Plug-in  
No Manual Configuration

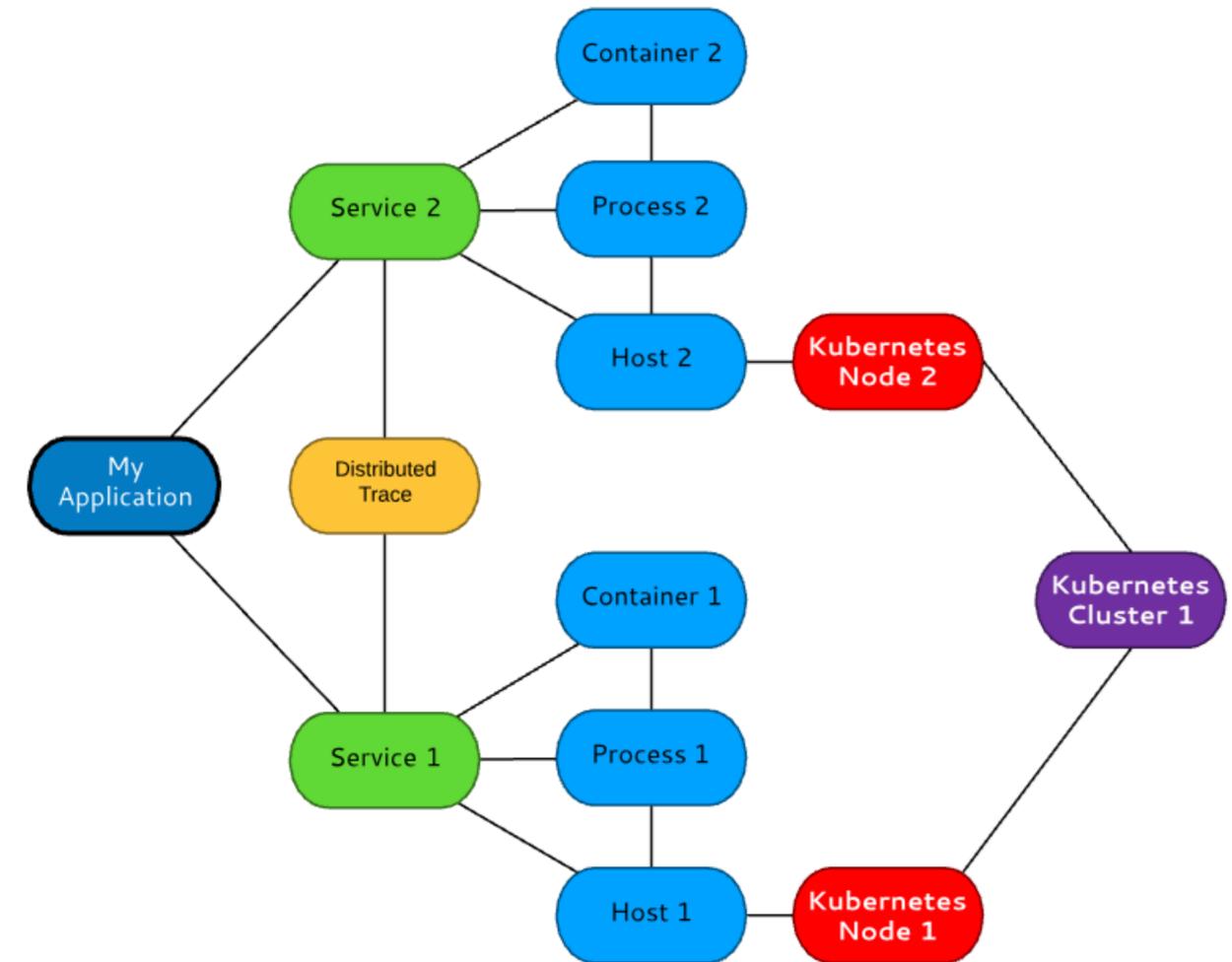
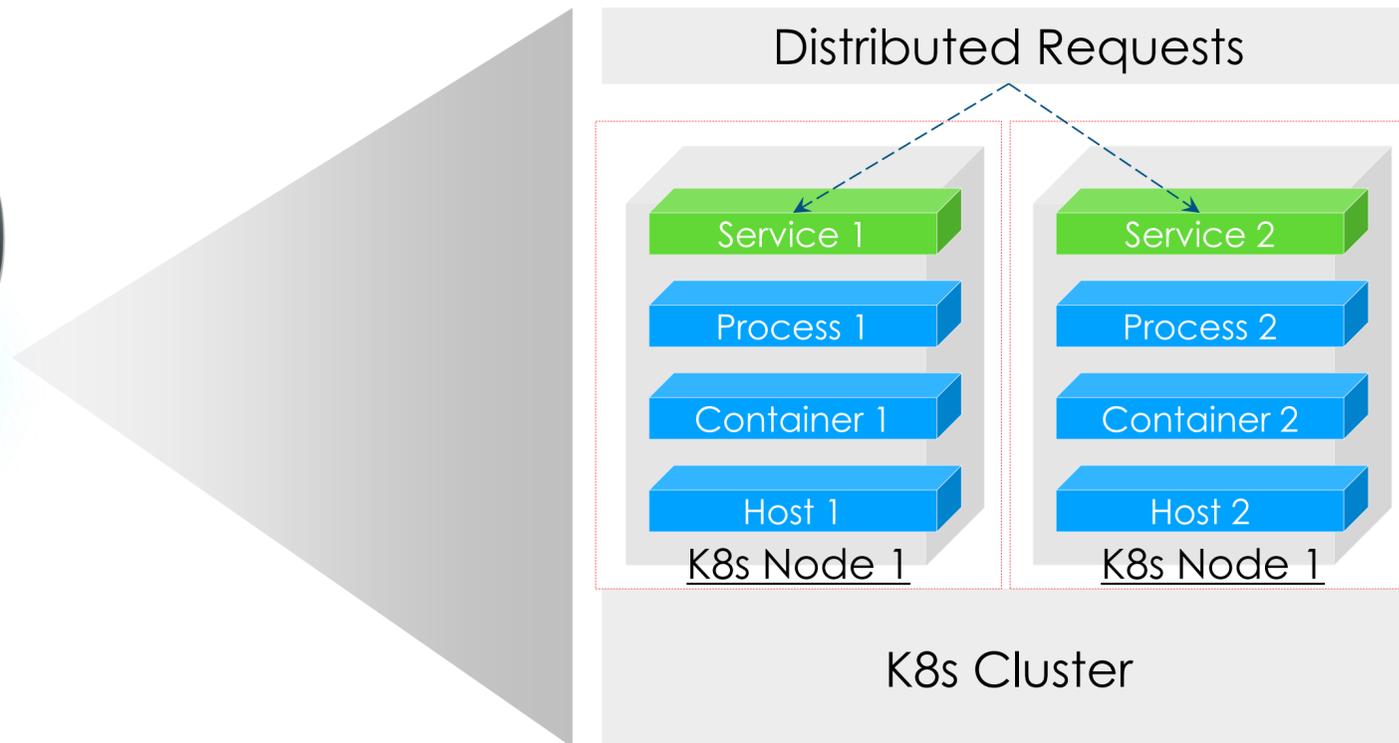


- Host(Server or VM)에서 수행되는 모든 사용자 request를 수집 (**Sampling 방식 아님**)
  - 동적인 Application 환경(Microservice)에서 발생하는 모든 정보 및 Issue를 놓치지 않고 실시간 빠짐없이 제공

모든 Request 정보를 실시간으로 수집하여, Microservice의 상태에 대한 정확한 정보 제공



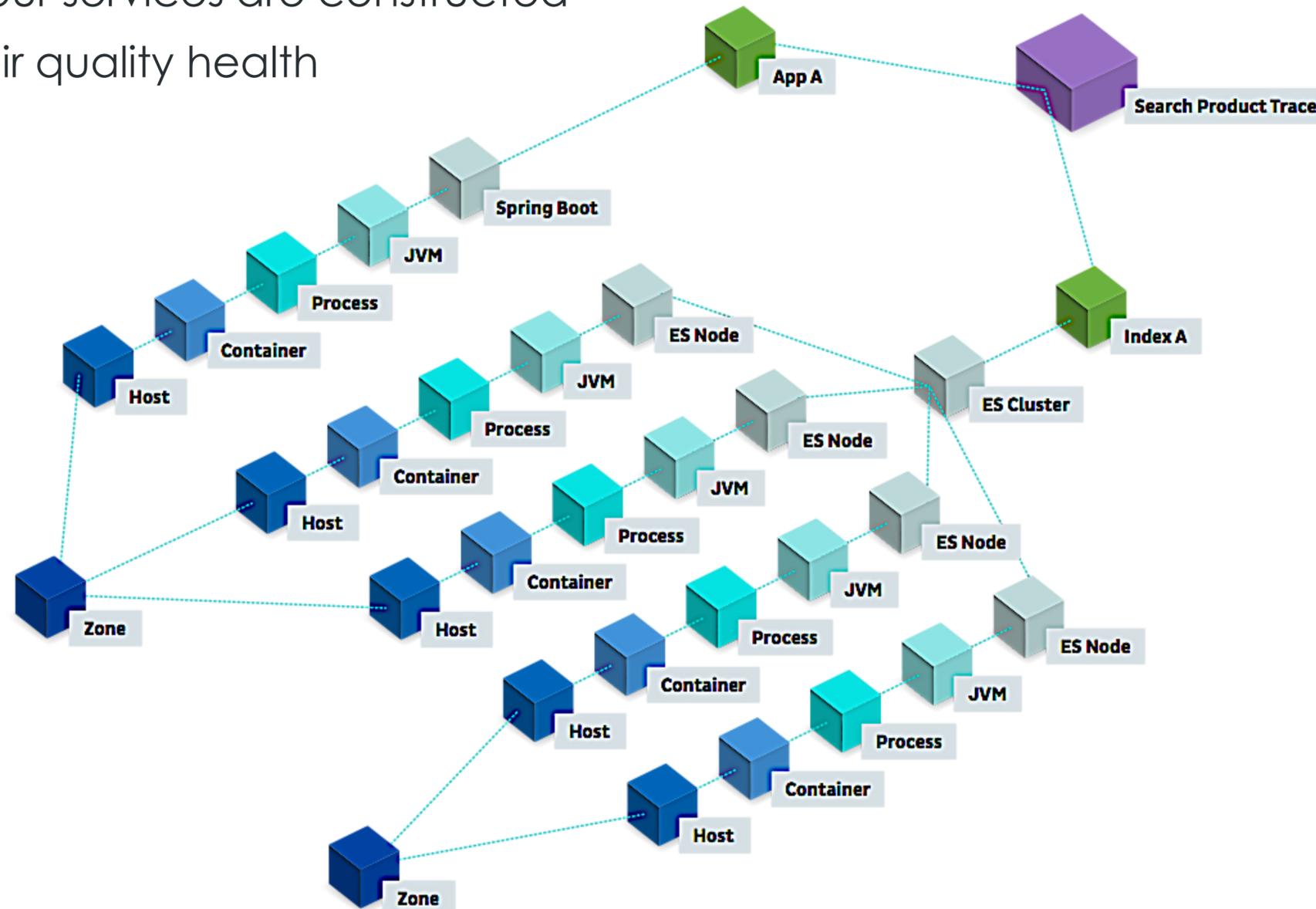
- 실시간으로 수집된 모든 Data정보를 기반으로, Application을 구성하는 모든 Component간의 물리적, 논리적 연관 관계를 이용하여 Dynamic Graph 생성
  - Dynamic Graph는 Machin Learning 기술을 활용하여, 시스템에 발생하는 Issue를 분석하고 Root Cause를 신속하게 제공



Component 연관 관계를 바탕으로 Dynamic Graph 생성 -----> 생성된 Dynamic Graph

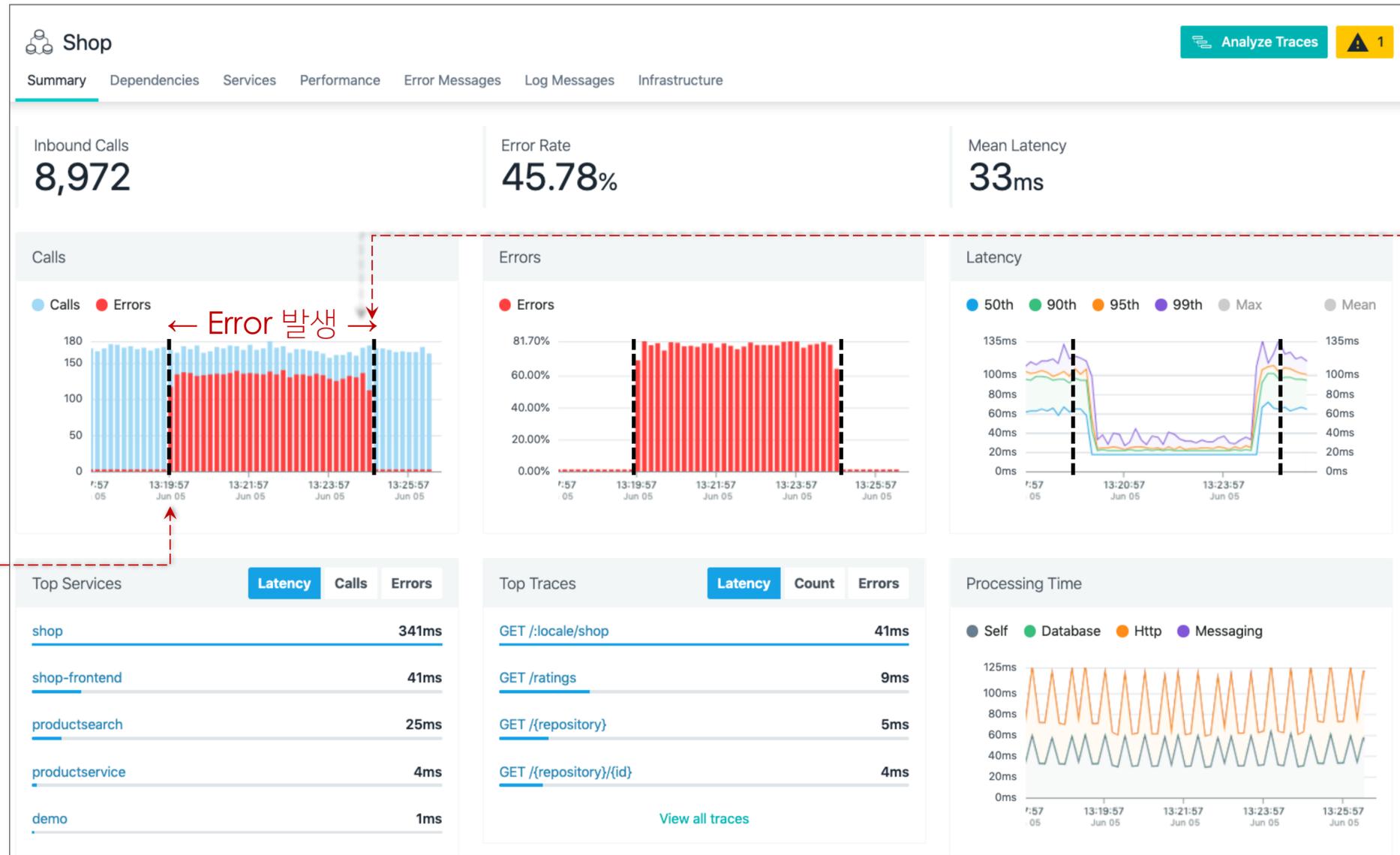
- The Dynamic Graph is Instana's internal data model

- What are your services
- How are your services are constructed
- What is their quality health



- Created and updated automatically
- Correlates services to infrastructure, process, and traces
- Dependency understanding enables intelligent analysis
- The Dynamic Graph keeps the history of your application's architecture

- 모든 Data 지표를 1초 단위로 수집하여, 서비스 Application의 새로운 Version 배포 시, 즉각적인 검증 정보 제공



새로운 서비스 배포 시점  
; New Version Release

배포시, Error발생으로  
인한 Rollback 시점

- 수집한 모든 Trace에 대해서 Instana가 제공하는 Unbounded Analyze를 이용하여, Issue가 있는 어떠한 Trace도 직관적으로 신속히 검색 가능
  - Instana가 제공하는 Trace에 대한 상세한 정보를 통해 Issue를 쉽게 분석

**Find any request using Unbounded Analytics™**

May 21 Last 6 hours Live

Download

application.name equals Shop

trace.latency > 2000

Grouped by trace.endpoint.name x Change Group

Result 1 Group

Group	Count ↓	Earliest Timestamp	Latency (mean)
GET /locale/shop	27,627	2019-05-21 10:39:24	2,575ms

Sub Calls: 10, Errors in Calls: 0, Latency: 3.12s

Colorize by Endpoint Technology

Service	Endpoint	Calls	Aggregated Time ↓	Errors
shop-frontend	GET /locale/shop	1	3,118ms	0
shop	GET /shop	1	3,118ms	0
stackoverflow.com	GET /questions	1	3,012ms	0
productsearch	GET /productsearch	1	91ms	0
productservice	GET /{(repository)/se	1	3ms	0

Prev 1/2 Next

Colorize by Endpoint Technology

GET /es/shop HTTP To GET /locale/shop of shop-frontend 3,118ms

GET /shop HTTP To GET /shop of shop 3,118ms

**Call Details**

GET /productsearch HTTP

Service To GET /productsearch of productsearch

Infrastructure Instana Demo - Product Search 0.0.1

**Caller Details**

Host ip-172-31-16-127.ec2.internal:85

Request Path /productsearch

URL http://ip-172-31-16-127.ec2.internal:85/productsearch

Parameters name=demoproduct2025984555

Method GET

Status Code 200 - OK

Content Length 194

**Callee Details**

Host ip-172-31-16-127.ec2.internal:85

Request Path /productsearch

Path Template /productsearch

Parameters name=demoproduct2025984555

Method GET

Status Code 200 - OK

Header: user-agent Java/1.8.0\_72-internal

Header: accept text/plain, application/json, application/\*+json, \*/\*

Caller Stack Trace

**Distributed Traces:** Instantly visualize the details of any request

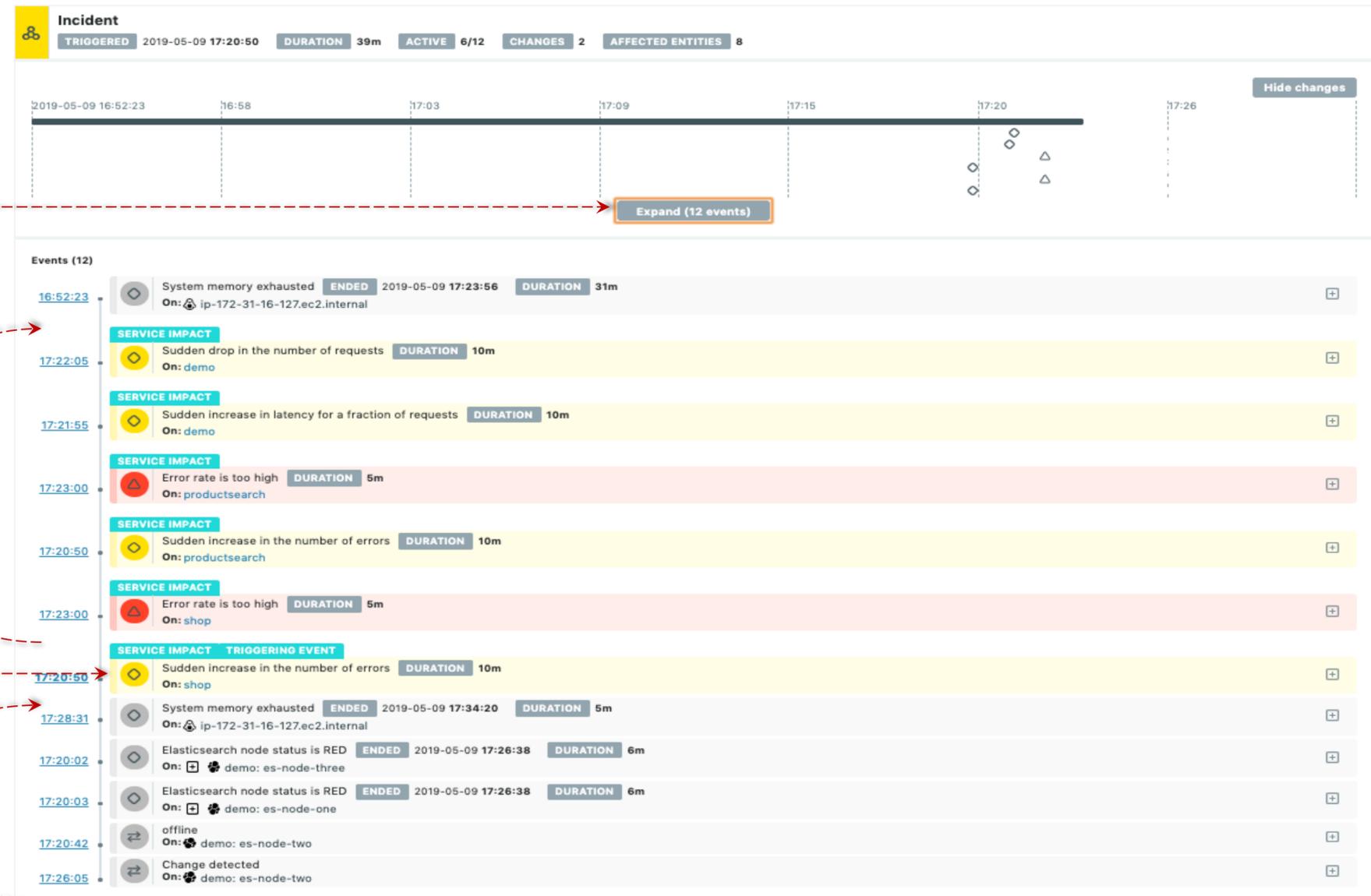
- 개별 Issue에 대해서 모든 Alert을 전송하지않고, 사용자가 사용하는 실제 서비스에 문제가 발생하는 Incident가 발생한 경우에 Alert 전송
  - Incident에는 해당 Issue와 관련된 모든 Component에서 발생한 정보를 함께 제공, Root Cause 및 관련 Issue를 신속히 처리

12개의 개별 alert을 하나의 연계 Event로 제공

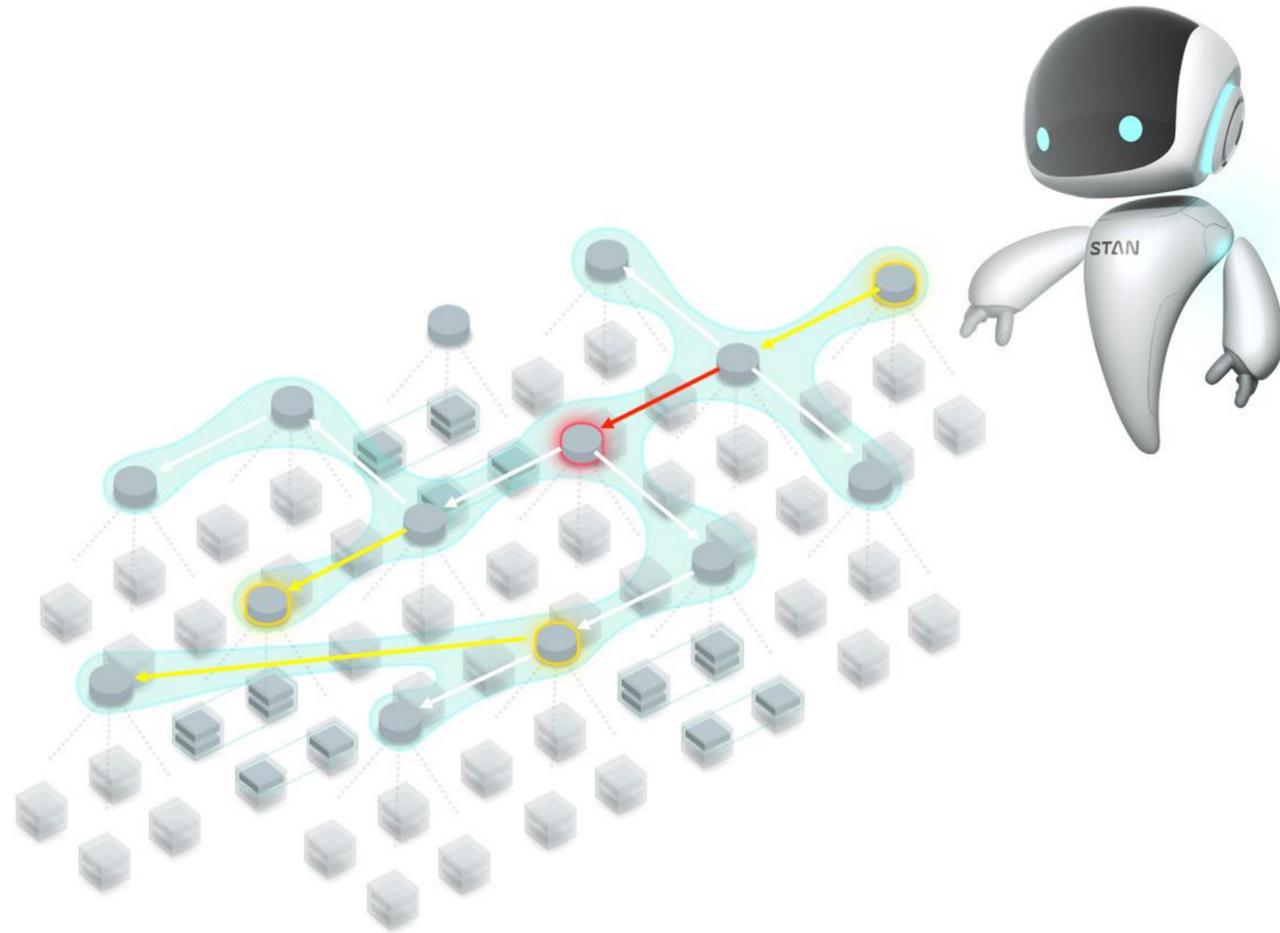
영향을 받은 서비스들

Event Trigger 서비스  
(가장 먼저 조치할 사항)

예상되는 원인들



# Instana 주요 기능



- Host/VM 단일 Agent

- Host(VM)당 한 개의 Agent설치만으로, Host내의 모든 Infrastructure에 대해서 Monitoring 수행

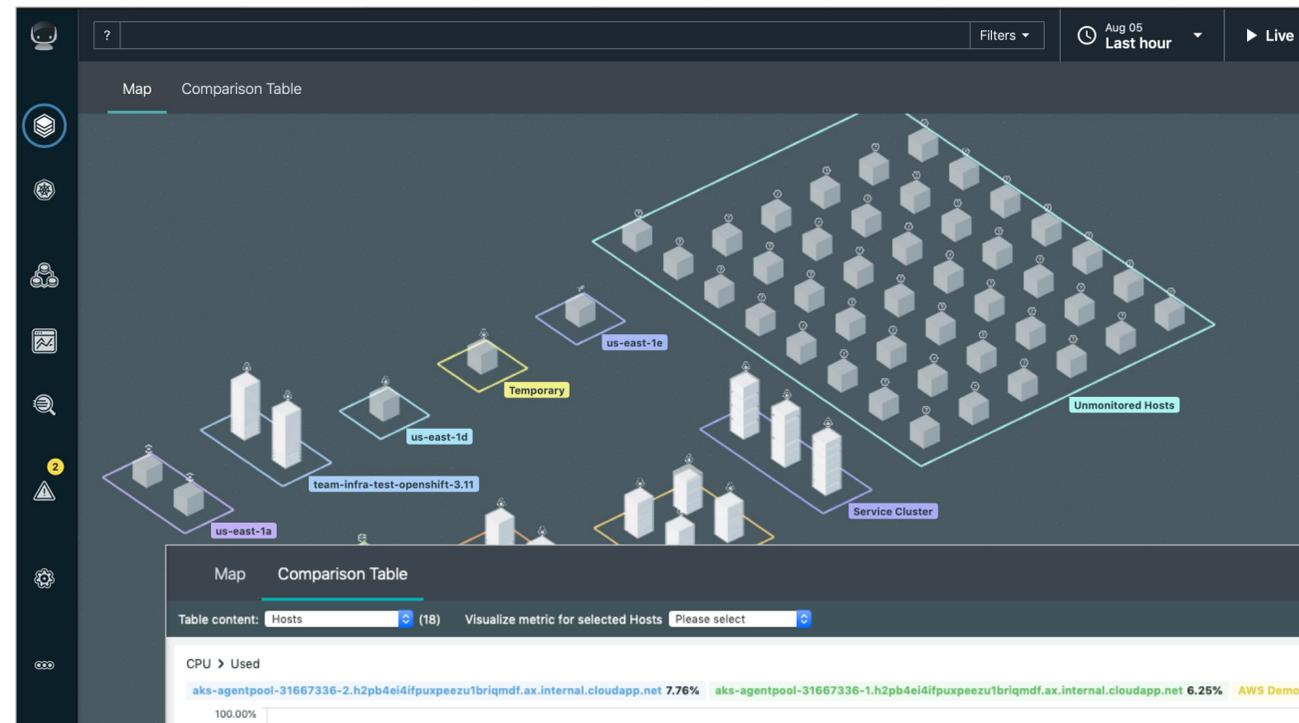
- Auto Discovery

- Host에서 실행, 추가, 변경되는 모든 Component를 자동으로 discovery하여 Monitoring 수행

- Infrastructure Map/Comparison Table

- 직관적인 가시성을 제공하는 Infra Map과 Component간 비교를 용이하게 해주는 테이블 형식의 Dashboard 지원

Infrastructure map



Comparison table

Map Comparison Table

Table content: Hosts (18) Visualize metric for selected Hosts Please select Aggregates for metrics over 1h 13m Clear Selections

CPU > Used

aks-agentpool-31667336-2.h2pb4ei4ifpuxpeezu1briqmdf.ax.internal.cloudapp.net 7.76% aks-agentpool-31667336-1.h2pb4ei4ifpuxpeezu1briqmdf.ax.internal.cloudapp.net 6.25% AWS Demo 10.93%

Zone	Name	Hostname	OS	Type	#CPUs	CPU Usage	Memory	Memory Used	Health
AKS-1-12-4-AW	aks-agentpool-31667336-0.h2pb4ei4ifpuxpeezu1briqmdf.ax.internal.cloudapp.net	aks-agentpool-31667336-0	4.15.0-1037-azure (amd64)	Standard_DS2_v2	2	6%	6.81 GiB	26%	✓
AKS-1-12-4-AW	aks-agentpool-31667336-1.h2pb4ei4ifpuxpeezu1briqmdf.ax.internal.cloudapp.net	aks-agentpool-31667336-1	4.15.0-1037-azure (amd64)	Standard_DS2_v2	2	6%	6.81 GiB	25%	✓
AKS-1-12-4-AW	aks-agentpool-31667336-2.h2pb4ei4ifpuxpeezu1briqmdf.ax.internal.cloudapp.net	aks-agentpool-31667336-2	4.15.0-1037-azure (amd64)	Standard_DS2_v2	2	8%	6.81 GiB	23%	✓
us-east-1d	AWS Demo	ip-172-31-85-122	4.15.0-1032-aws (amd64)	t2.small	1	11%	1.95 GiB	36%	✓
k8s-demo	gke-instana-test-default-pool-a6635db3-09pr.c.k8s-brewery.internal	gke-instana-test-default-pool-a6635db3-09pr	4.14.106+ (amd64)	n1-standard-2	2	10%	7.30 GiB	61%	✓

## ■ Component별 개별 Dashboard

- 개별 기술 Component에 맞는 지표 포함, 사전 정의된 Dashboard 제공

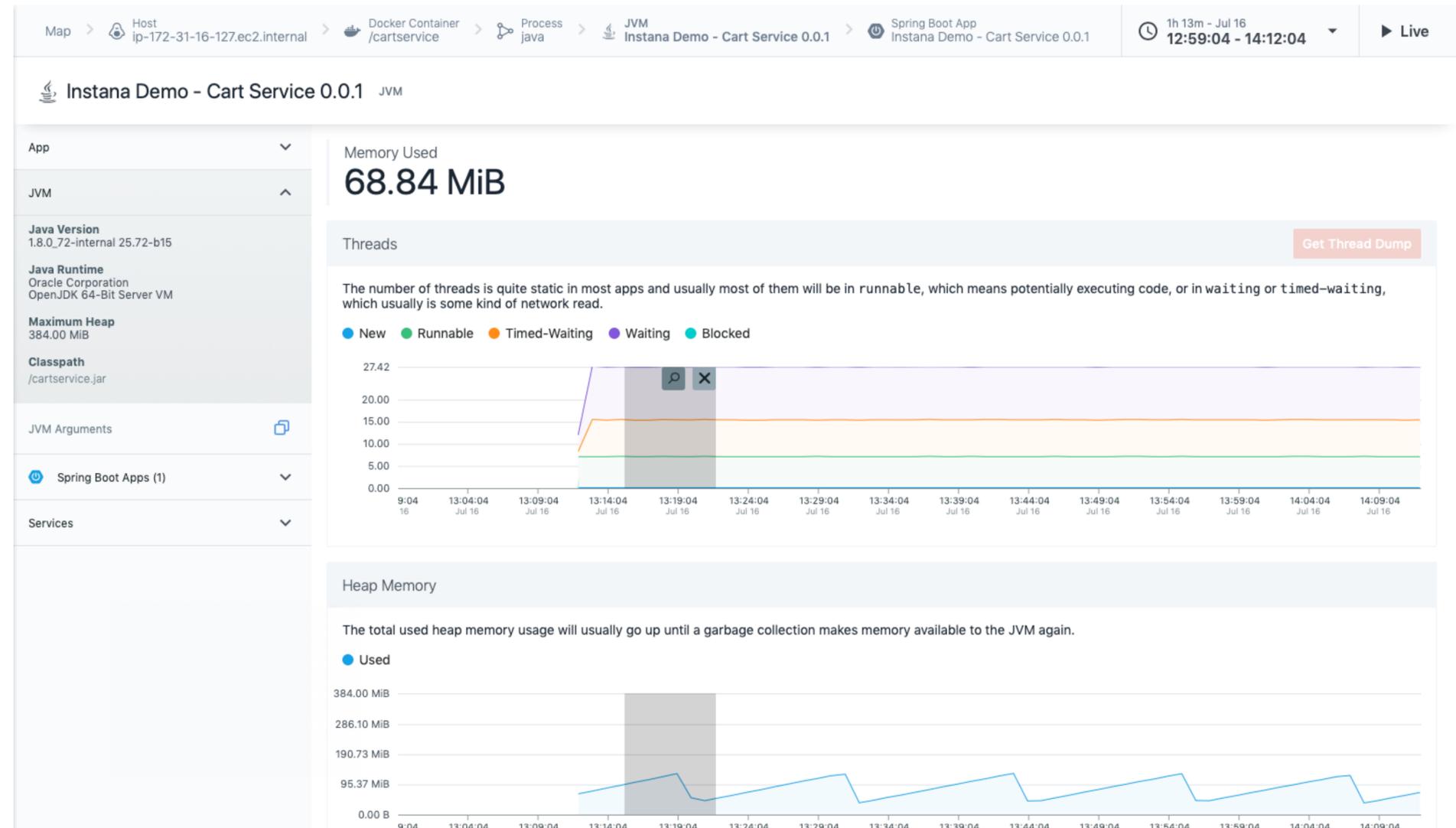
### Component별 개별 Dashboard

## ■ 1초 단위 지표 제공

- 1초 단위 지표 제공을 통해 동적인 Application 환경에 대응

## ■ Dynamic Query

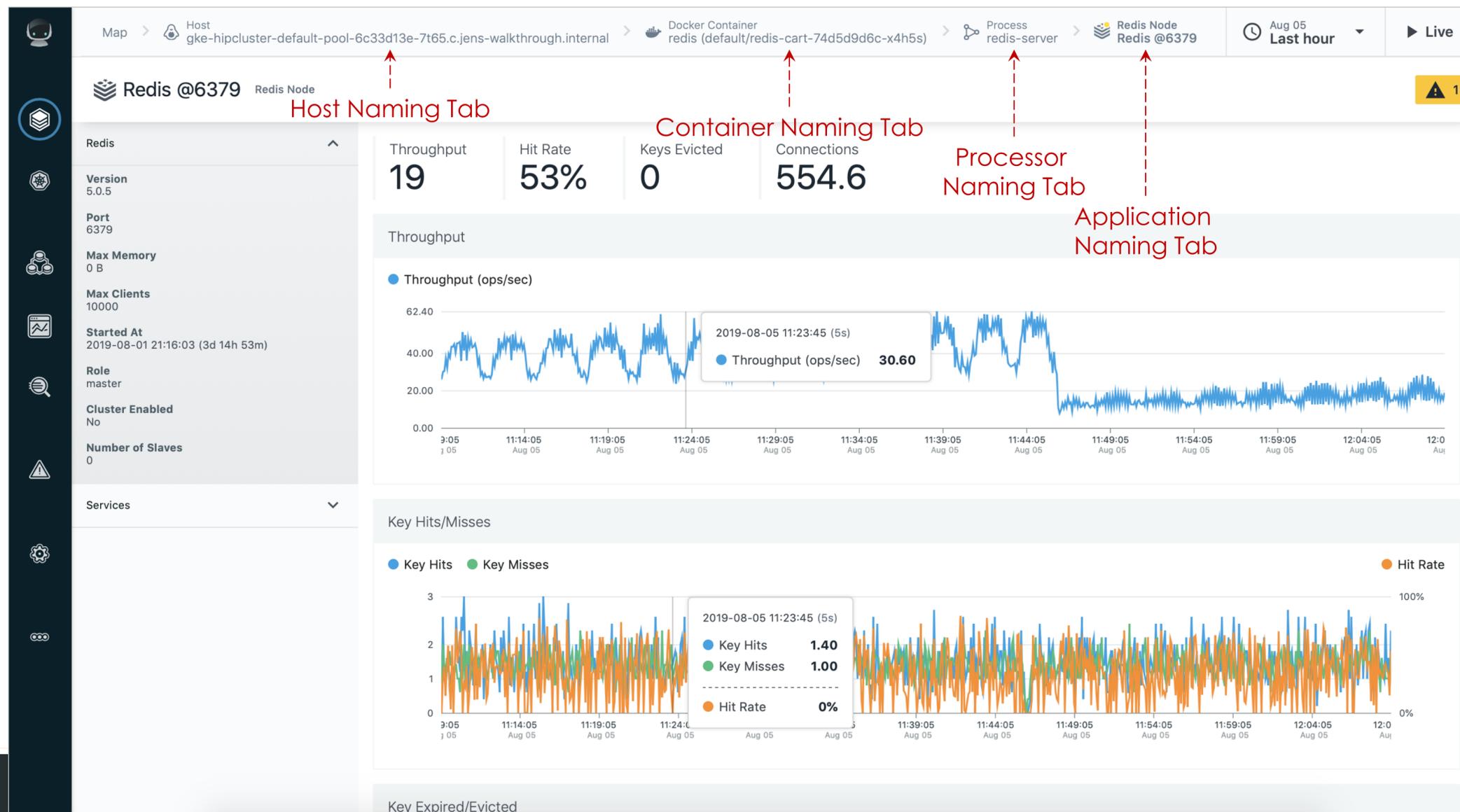
- Lucene Query와 Tag를 통해 원하는 Host 혹은 Component만 검색 가능



## Dynamic Graph

- Host, Container, Processor, Application 등, Component간의 연관관계를 Dynamic Graph기반으로 분석하며, 연계된 서비스와 Trace 정보를 제공

### Component들간의 연관관계 분석한 Dynamic Graph Tab View



- Kubernetes 구성 정보 및 Monitoring
  - Cluster와 Namespace를 구성하는 다양한 Component 단위의 구성정보, 성능지표, Event 등 Monitoring

- Resource Monitoring

- Cluster와 Namespace등에 할당된 Resource 정보 제공

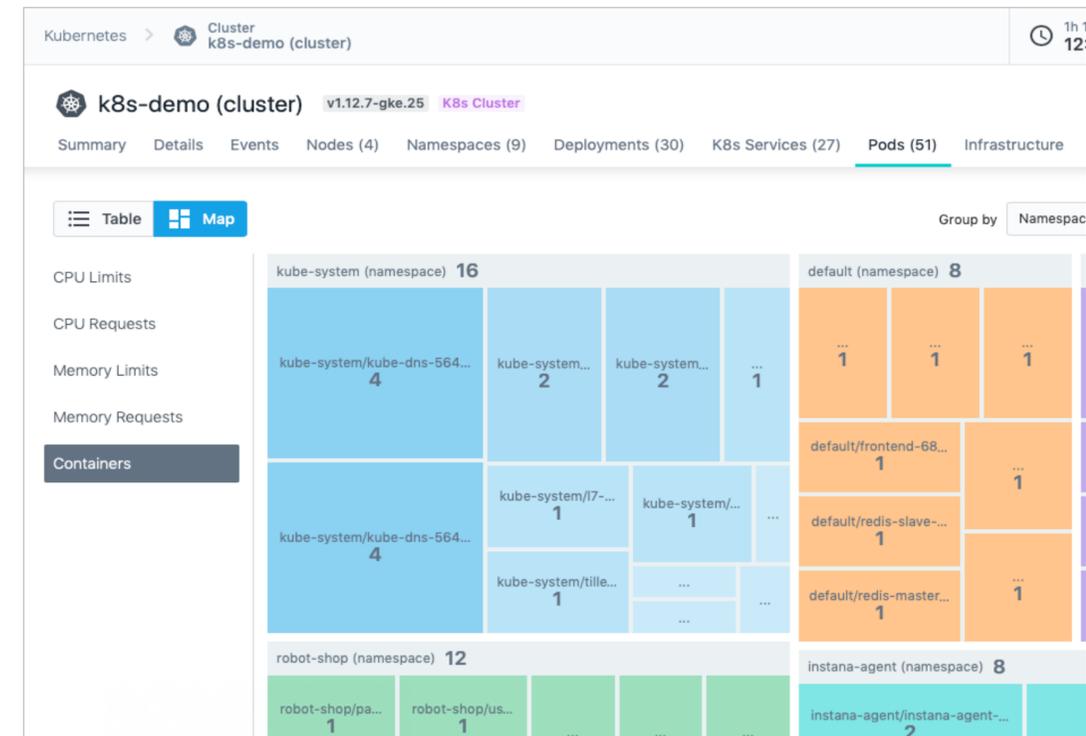
- 서비스와 연계

- Cluster, Namespace, Deployment, Port에서 실행되는 서비스를 자동 Discovery 하고, 실행되는 Tracing 정보를 해당 Component 단위로 직접 조회

## Cluster Dashboard

Name ↑	Namespaces	Services	Nodes	Pods	Depl
AKS-1-12-4-AW (cluster)	4	8	3	28	
k8s-demo (cluster)	9	27	4	51	
team-infra-test-openshift-3.11 (cluster)	19	32	3	329	

## Pods Map



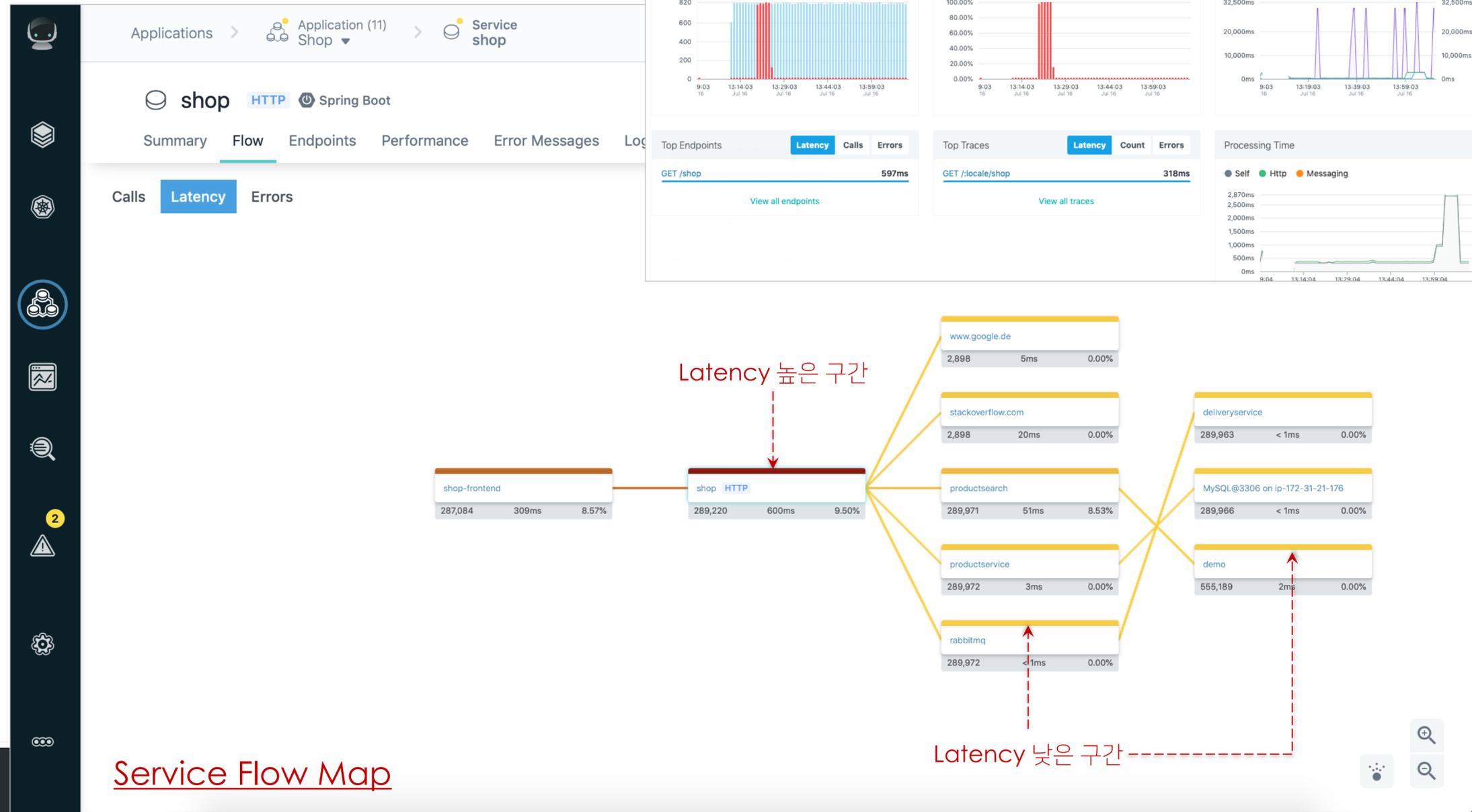
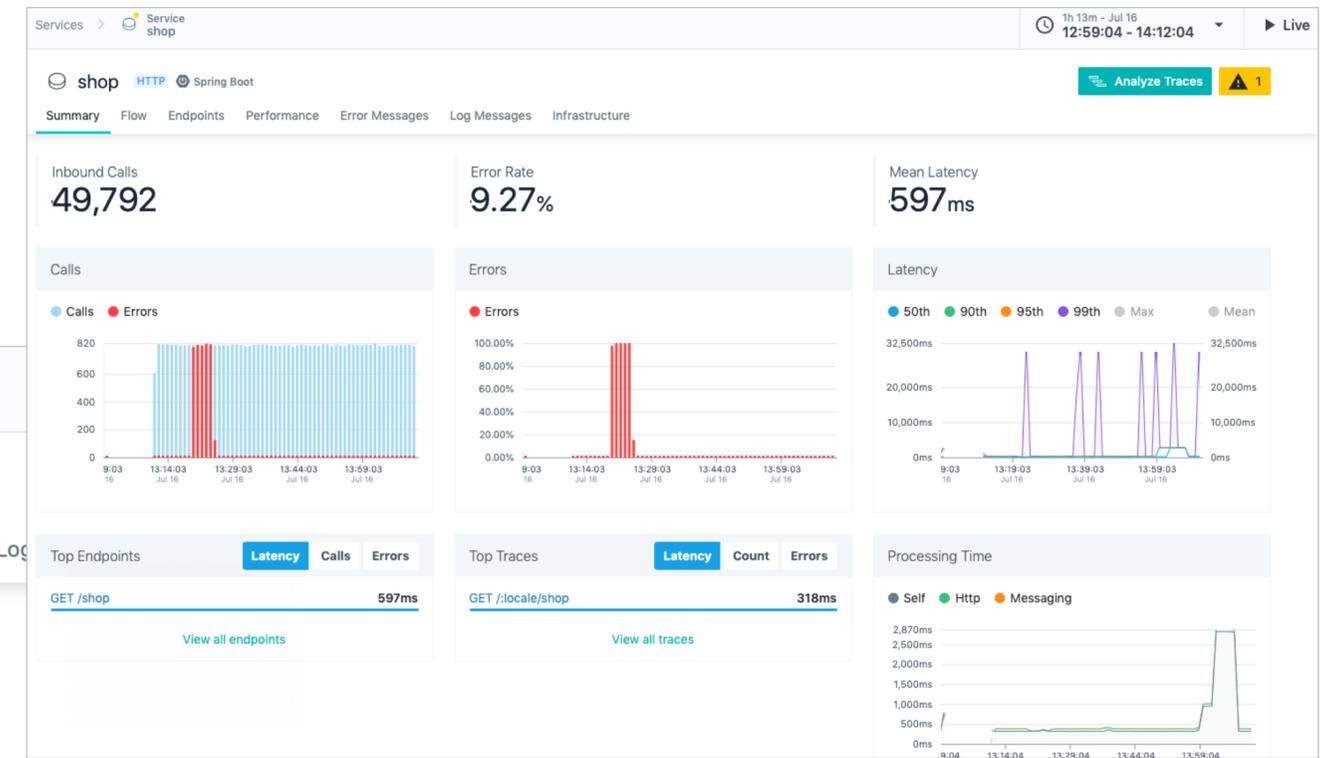
- Infrastructure와 연계
  - Cluster를 구성하는 Component와 연계되는 Infrastructure 정보 제공
  
- Red hat / OpenShift / Cloud Foundry 지원
  
- Managed Cloud 서비스 지원
  - Google K8S Engine, Azure K8S Service, AWS Elastic K8S Service 등의 다양한 관리형 클라우드에서 실행되는 Kubernetes 환경 지원

- 서비스 및 Endpoint 자동 분석
  - 서비스간의 호출 데이터를 바탕으로 서비스와 endpoint를 자동으로 분석 추출

- 서비스 Flow Map
  - 서비스간의 호출 관계를 알 수 있는 Flow Map을 자동으로 생성 제공

- Application 관점 View
  - 서비스를 사용자가 원하는 방식으로 볼 수 있는 Application 관점에서 관리

## Service Dashboard

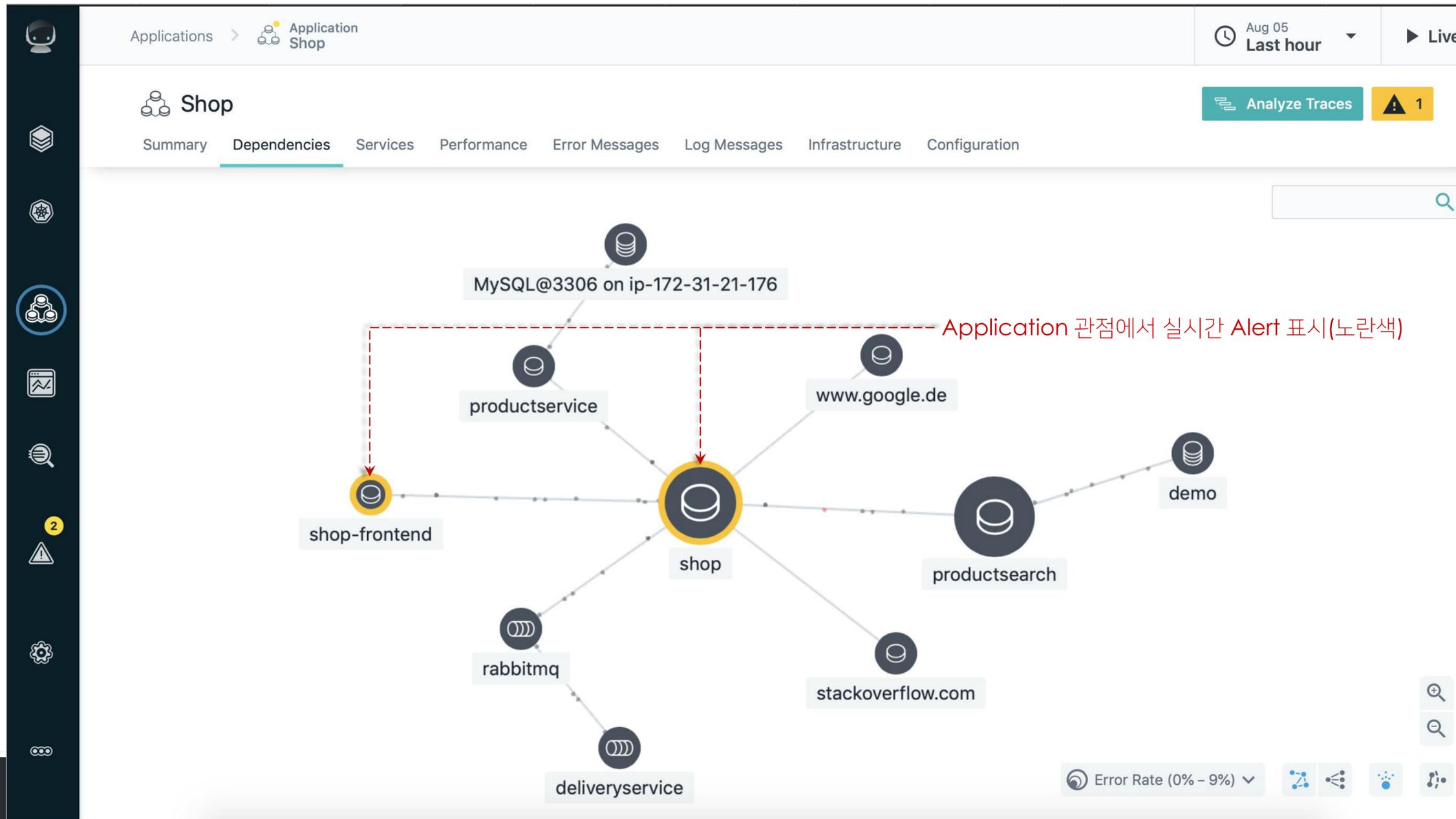


Service Flow Map

## ■ 서비스 연계 분석

- Application을 구성하는 서비스들의 연관관계를 한눈에 파악할 수 있는 Dependency Map 자동 생성

### Service Dependency Map



- Google Dapper 방법론 적용
  - 각 서비스/Application Dashboard google Dapper 방법론에 따라 호출수, Error율, 응답시간을 주요 지표로 제공
- Log/Error Monitoring
  - 서비스 수행시 발생하는 Log/Error 정보가 있는 Log파일을 검색하지 않고 Logging Framework을 Monitoring
- Trace 분석 연계
  - 개별 서비스 혹은 Application에서 수행된 Trace 분석 Page로 즉시 이동하여 효과적인 직관성 제공
- Synthetic 호출 제외
  - Health Check와 같은 인위적인 서비스 호출은 성능 지표에서 제외, 정확히 필요한 정보만 제공

## ■ 모든 Page Load 수집

- Browser에서 실행된 모든 Page Load를 Sampling없이 100% 수집하여 성능 분석 정보 제공

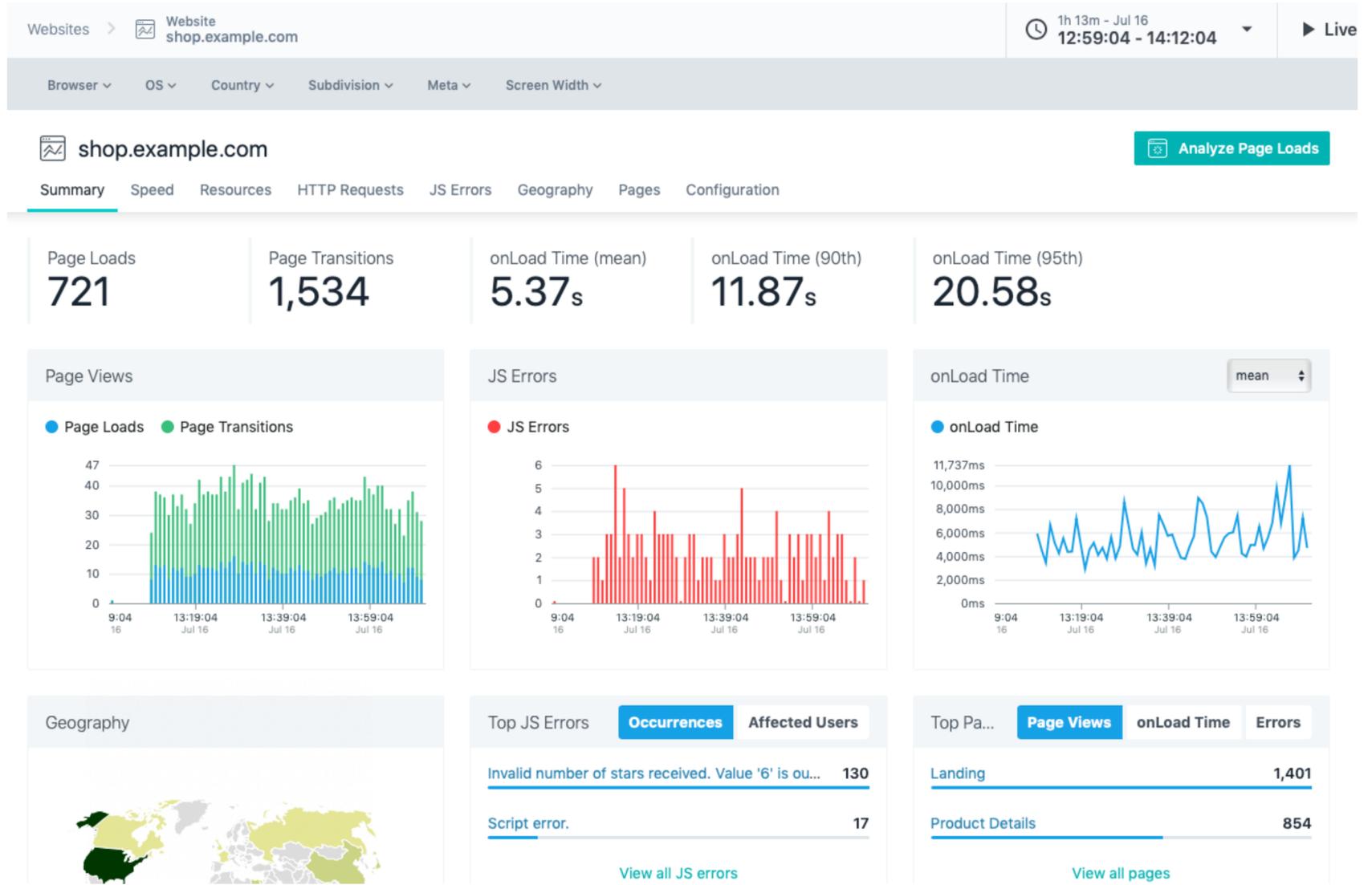
### Websites Dashboard

## ■ 상세한 성능 정보

- Page Load 횟수, Load 시간, Page 전환 등의 지표 및 Page Load에 소요된 개별 작업 소요 시간 정보 및 분석 정보 제공

## ■ 다양한 필터 제공

- Browser 종류, 사용자 운영 OS, 사용자 위치 등의 필터 제공



- Page 단위 성능 정보 확인

- Page Load에 필요한 개별 Resource에 대한 성능 정보 제공

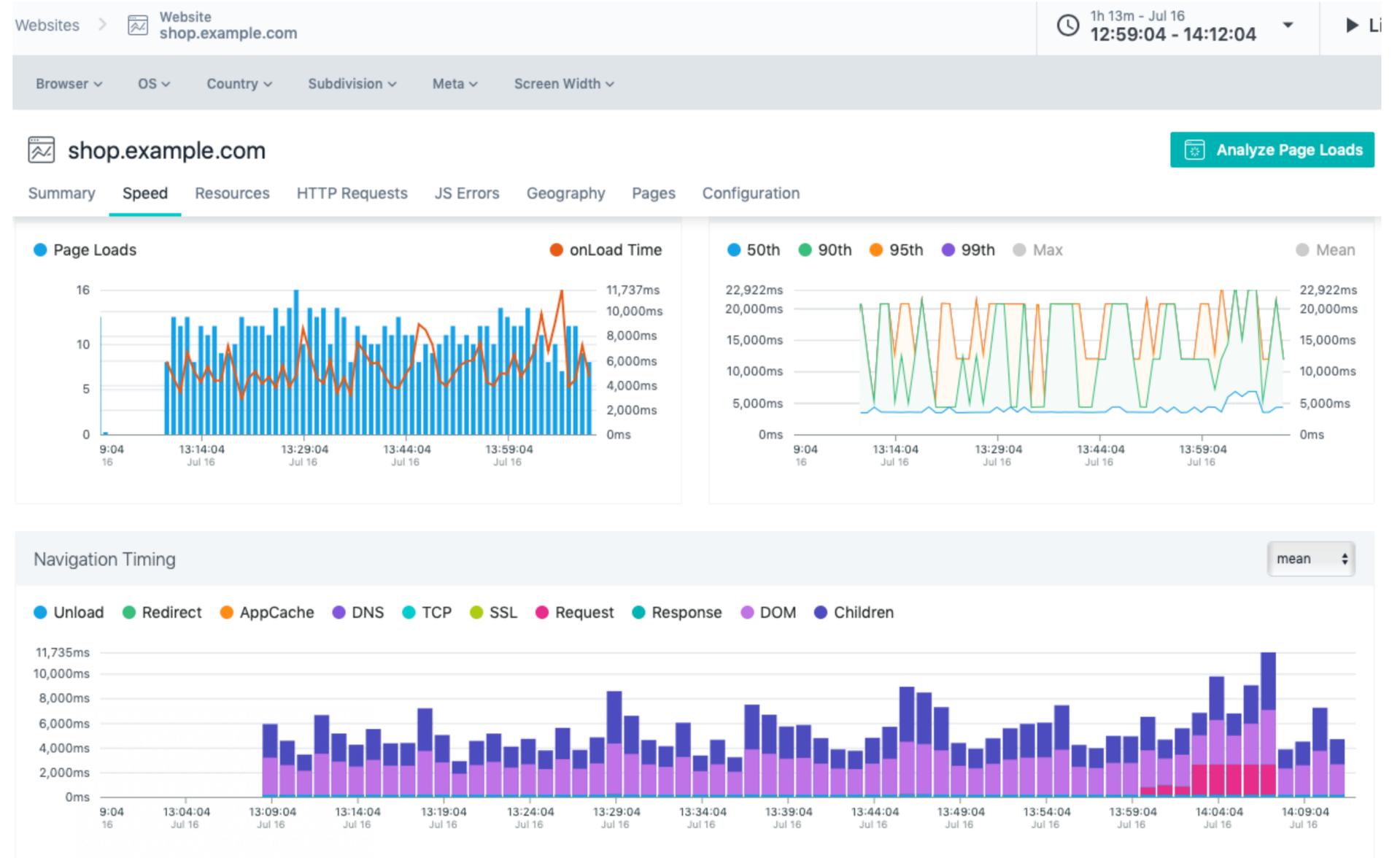
## Websites 성능 Dashboard

- Meta Data 수집 기능

- 사용자의 위치, 사용자 정보, Browser 정보 등의 다양한 Meta Data 수집

- Trace 분석 연계

- Dashboard에서 Page 단위의 Page Load 분석 Page로 즉시 이동 가능



- 모든 Trace / Page Load 수집

- 모든 Trace, Page Load 정보 빠짐 없이 수집 분석

- Code 변경 없는 Trace 수집

- Trace 수집을 위해 Code 추가나 변경, 재시작 작업 불필요

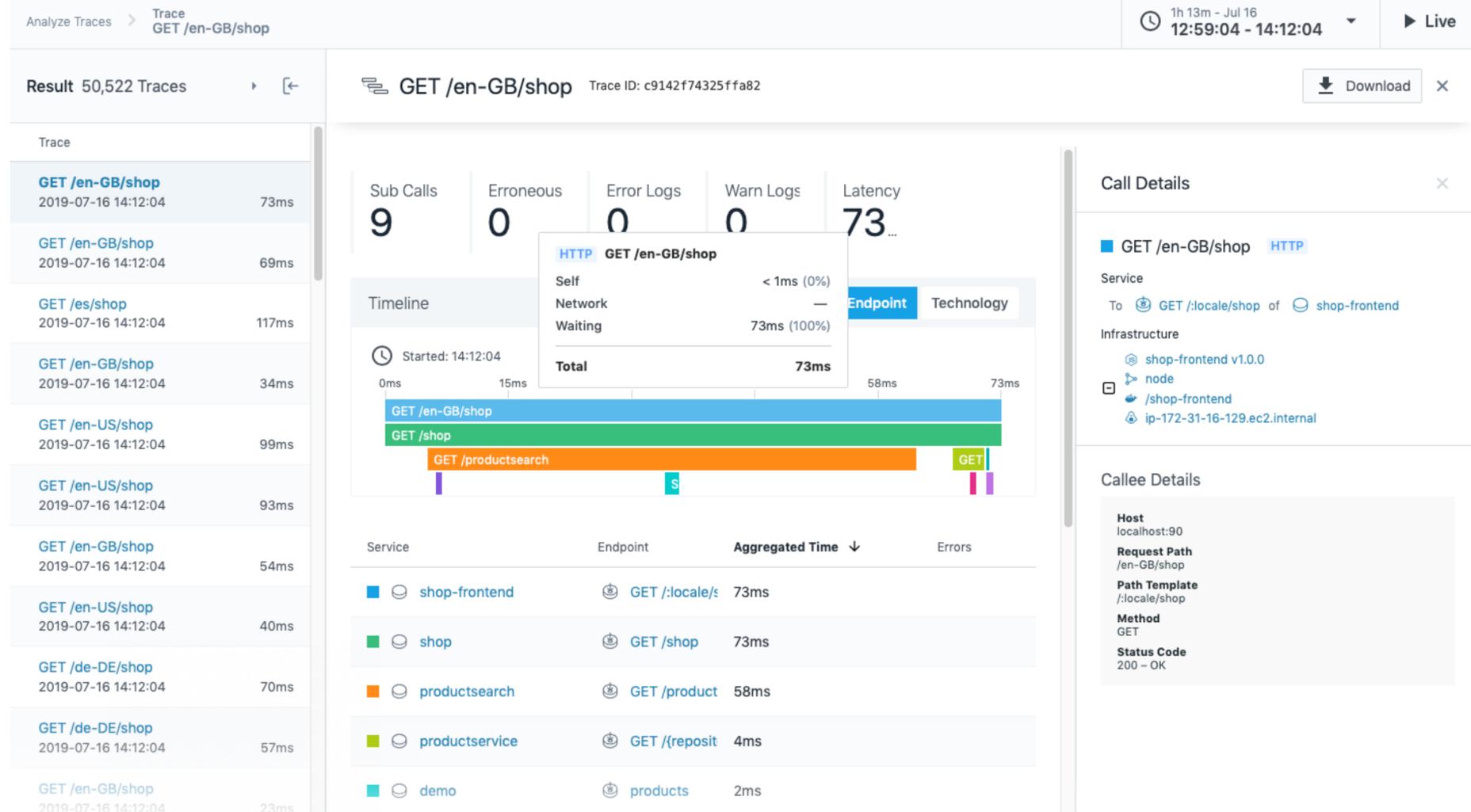
- Infrastructure, Service, Cluster와 연계

- 해당 Trace가 실행된 모든 Component와 연계 정보 제공

- End to End 추적 지원

- 모든 Trace/Page Load 수집을 바탕으로 Browser에서 Backend까지의 End to End 추적 지원하며, 오류 혹은 서비스 처리 지연 등을 확인한 경우, 서비스가 실행된 특정 하드웨어 Component를 즉시 특정하여 확인

## Trace 정보



- 도식화된 View 제공

- 개별 Trace 단위로 하위 서비스 호출을 Timeline 및 Tree 형식으로 도식화 제공
- 각 서비스 호출에 대해서 해당 Call이 어느 환경에서 실행되었는지 알 수 있는 상세한 정보 제공

- Unbounded Analyze

- 다양한 Filter와 Grouping 방법을 제공하는 Unbounded Analyze를 통해 어떠한 Trace도 빠르고 쉽게 찾아서 분석 가능

서비스 호출에 대한 상세 정보

The screenshot shows a trace entry for a MySQL call. At the top, it indicates the call is from 'prod...' to 'MySQL@3306 on ip-172-31-2...'. Below this, the 'Infrastructure' section shows the service as 'MySQL @3306'. The 'Caller Details' section contains the following information:

- Connection:** jdbc:mysql://ip-172-31-21-176/productdb
- Statement:**

```
1 | SELECT product0_.id as id1_0_, product0_
2 | FROM product product0_
3 | WHERE product0_.name is null
```

## ■ 사전 정의된 Event 제공

- 각 Component에 대해서 사전에 정의된 수 백 개의 Event 제공, 수집된 지표를 기반으로 사용자 정의 Event 생성

## ■ Changes

- Process/Server 등의 시작 종료 등의 상태 변화 기록 관리하며, 속성, 버전 등의 변경 기록 관리. 변경 시점, 변경 위치, 변경 내용 수집

## ■ Issues

- 정의된 Event가 발생시, 발생 당시의 지표 제공

## ■ Incidents

- 개별 Component의 Issue를 종합하여, 서비스 문제가 발생한 시점에서 발생하는 Event로 해당 Incident에 영향을 준 Issue들을 영향을 받은 Component 정보를 함께 제공

## ■ 다양한 Alert 채널 지원

- Slack 등 다양한 Alert 채널 제공

### 변경 상세 내용

Refresh every 10 seconds

**Change detected**

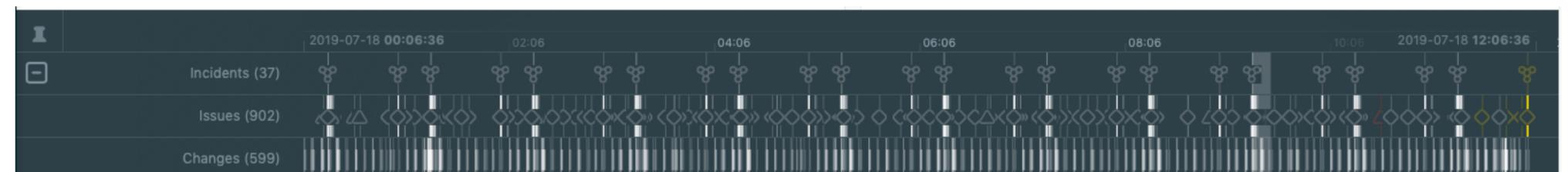
On: demo: es-node-two  
Elasticsearch: Core 2.3.4  
/opt/oracle/jdk1.8.0\_71/bin/java  
ip-172-31-27-58.ec2.internal

TIME 2019-07-18 11:44:07

**Detail**

- The value **index.replicas** has changed:
- The value **index.shards** has changed:
- The value **index.names** has changed: values [ "products" ] added.

### Event Timeline



# INSTANA

감사합니다.

