Application Modernization in the Cloud Age

INSTANA
an IBM Company

Tom Fisher

Application Modernization

Why it's Occurring and is Important

Business Agility

Ability to bring new application functionality online in real time

Bring new initiatives on line quickly

Adjust rapidly to changes in the competitive landscape

Responsiveness

Immediately respond to user concerns

Rapidly diagnose and remediate complex issues

Maintain SLI, SLO, and SLA goals

Scalability

Scale-up instantly as application demand peaks

Scale-down as application demand ebbs

Cost Effectiveness

Add and use resource credits based upon application scaling requirements

Reduce or eliminate application downtime by applying on-demand resources

The Evolution of Application Architectures

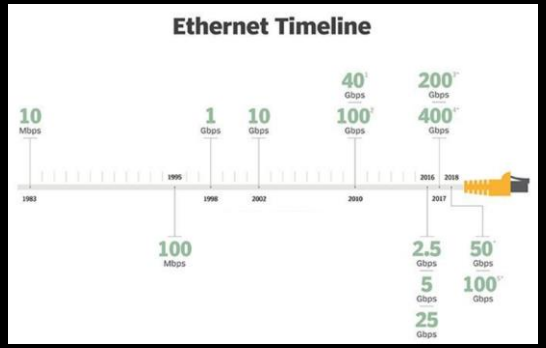
- Monolithic
  - 1948-1997
- SOA
  - 1998-2011
- VMs
  - 1972 – IBM
  - 1999 – VMware
- Microservices
  - 2011-

- Waterfall Design
  - 1970
- Agile
  - 2000

Why Application Architectures Evolved

- Mobile Internet
- Faster and more reliable backbone networks
  - Highly distributed services became much more viable
- Compiled to JIT VM code
  - Smalltalk ~ 80s
  - First JVM – 1994
- Faster processors, multi-core, denser storage, etc.

Mobile Network	Average Speed	Peak Speed
2G	0.1Mbps	0.3Mbps
3G	3Mbps	7.2Mbps
3G (HSPA+)	6Mbps	42Mbps
4G LTE	20Mbps	150Mbps
4G LTE Advanced	42Mbps	1Gbps
5G	500-700Mbps	10 or 20Gbps



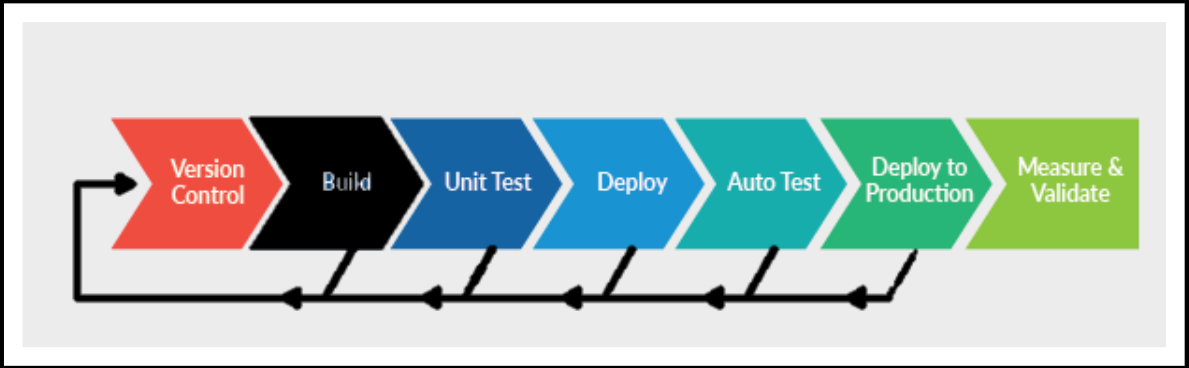
The 5 most common Application Modernization Options

Rehost	Move applications to cloud Infrastructure as a Service (IaaS) without altering their architecture
Refactor	Change the application code to fit a Platform-as-a-Service (PaaS) model
Rearchitect	Modify or extend the existing application code to become cloud-native
Rebuild	Rebuild application on PaaS, remove code for the existing platform, and rearchitect to take full advantage of cloud-native features
Replace	Replace existing application with a commercial Software-as-a-Service (SaaS) application

Where Enterprise Observability Helps Application Modernization

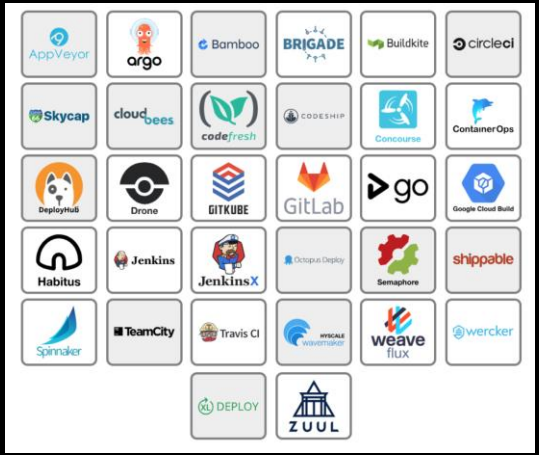
Application Modernization Option	Cloud Service Type						Cloud Configuration Type					
	IaaS		PaaS		SaaS		Single Cloud		Multi-Cloud		Hybrid-Cloud	
Rehost	X						X		X			
Refactor			X				X		X		X	
Rearchitect	X		X				X		X		X	
Rebuild			X				X		X		X	
Replace					X		X		X		X	

The CI/CD Pipeline



CI/CD Tools

Enable automation and monitoring for apps dev, integration and testing to deployment



Observability's Role in CI/CD Pipeline Optimization

Discover and address 'unknown unknowns'

Issues you don't know exist

Catch and resolve issues early in development

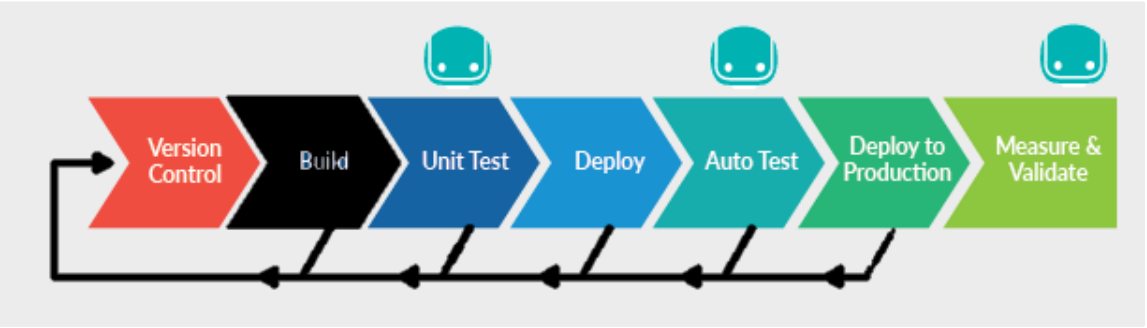
Automatically scale observability

Enable automated remediation and self-diagnosing application infrastructure

Shift-Left Observability

For Building Better Software Faster by

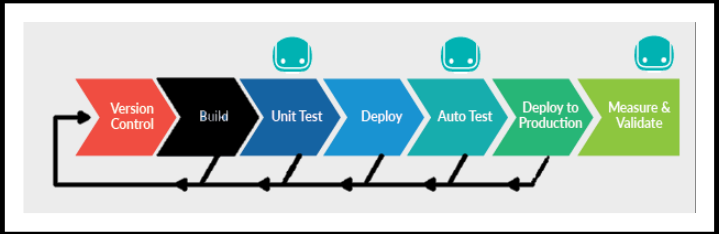
***Optimizing* Unit Test and Auto Test**



And of Course, for Production

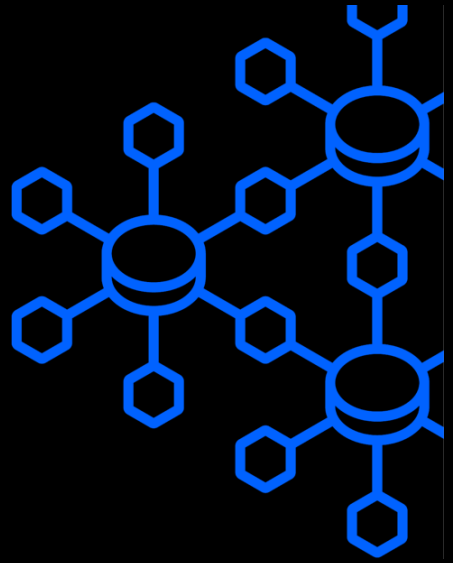
Optimizing the CI/CD Pipeline

- Unit Test Values
  - Automated Profiling provides code level details for triage

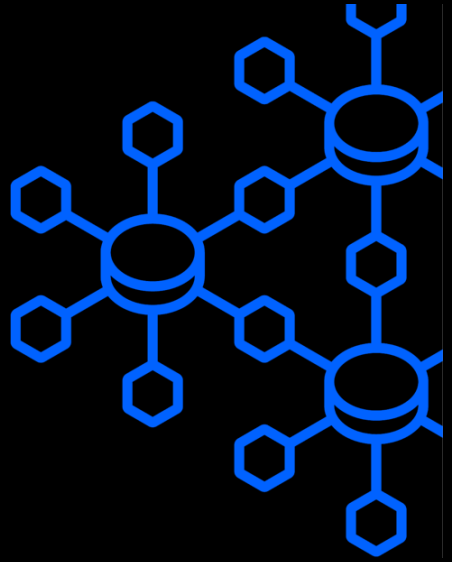


- Auto Test (and Production) Values
  - Automation
    - Discovers/maps apps, services, infrastructures, events, and dependencies
  - Context
    - Ingests all observability metrics, traces each request, profiles every process and updates dependency maps in real time
  - Intelligent Action
    - Machine Learning Analytics for optimizing application performance

Key Enterprise Observability Capabilities for Application Modernization



Automation

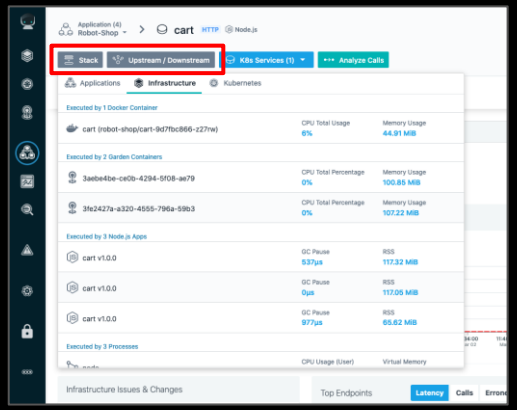


Automatic Profiling

- Automatic and continuous code level profiling
  - JVM, PHP, NodeJS, etc. tracers
- Profiler Sensor from the Agent
  - Always on, but not profiling all the time

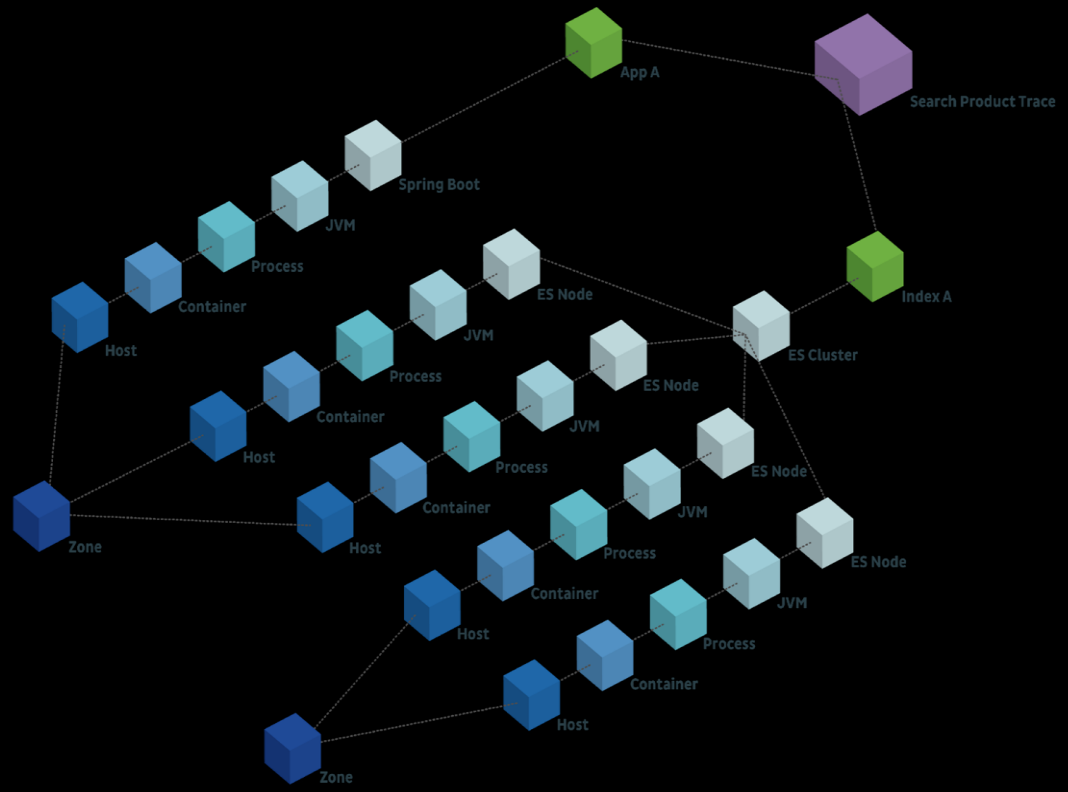
Enterprise Observability Automation

Immediate	
Exact	
Effortless	



Dynamic Graph

Continuously updated, full stack, internal data model of application structure and dependencies

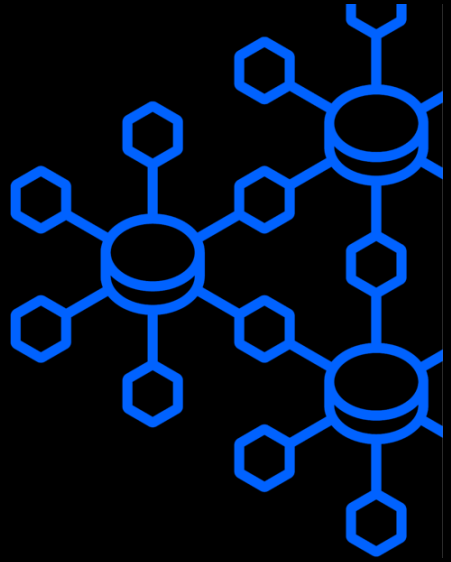


Context Guide

Dynamic Graph for rapid troubleshooting.

A “GPS” for enterprise applications.

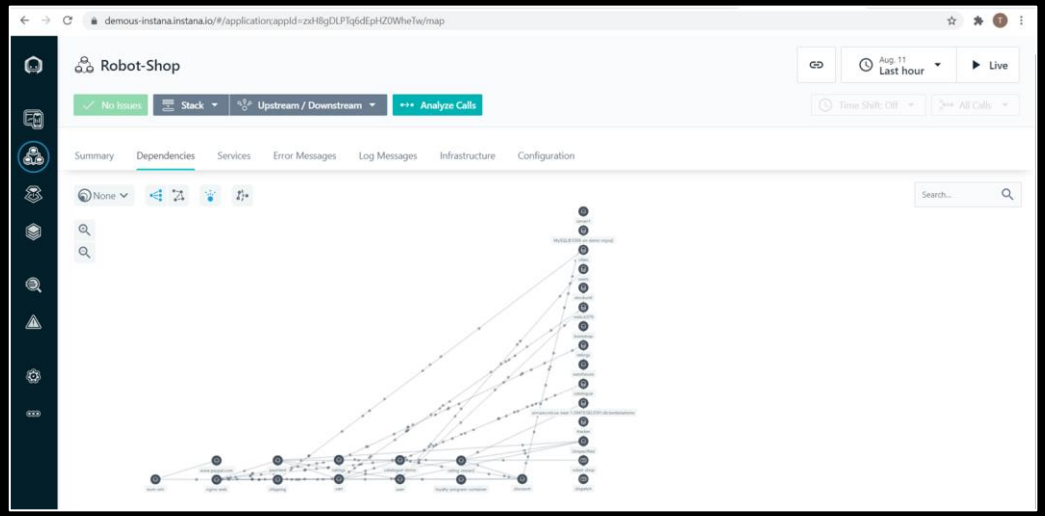
Context



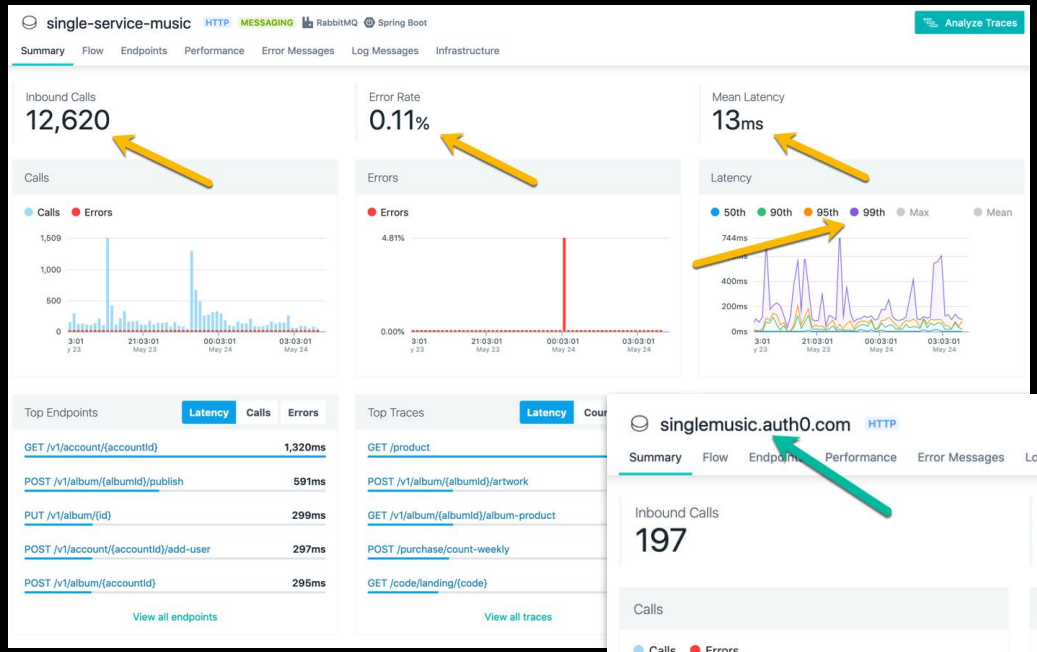
Application Dependency Maps

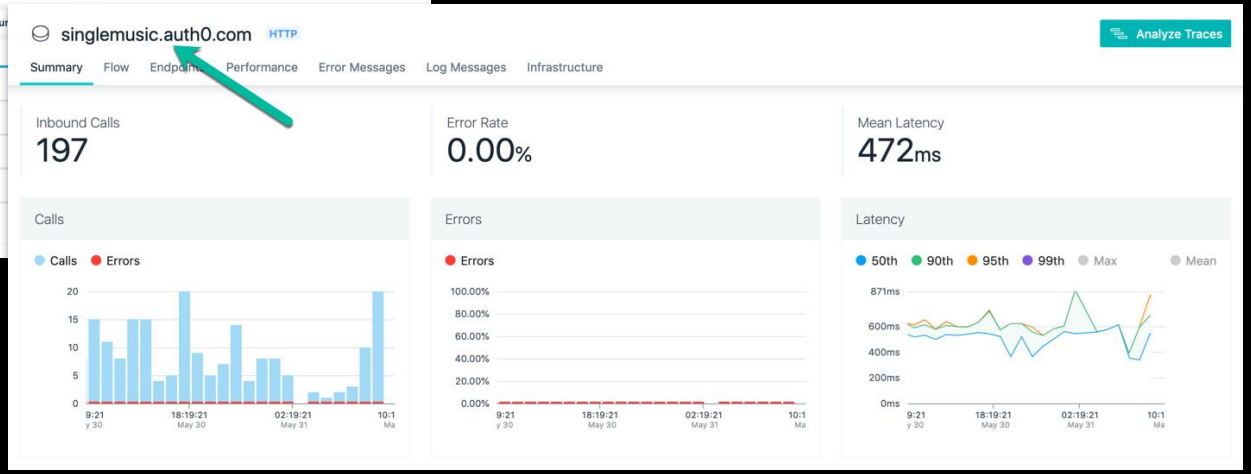
For Each Application

- Application service dependencies
- Calls between services
- Application architecture layout view
- Dashboards, flows, calls and issues service views



Golden Signals for All Services

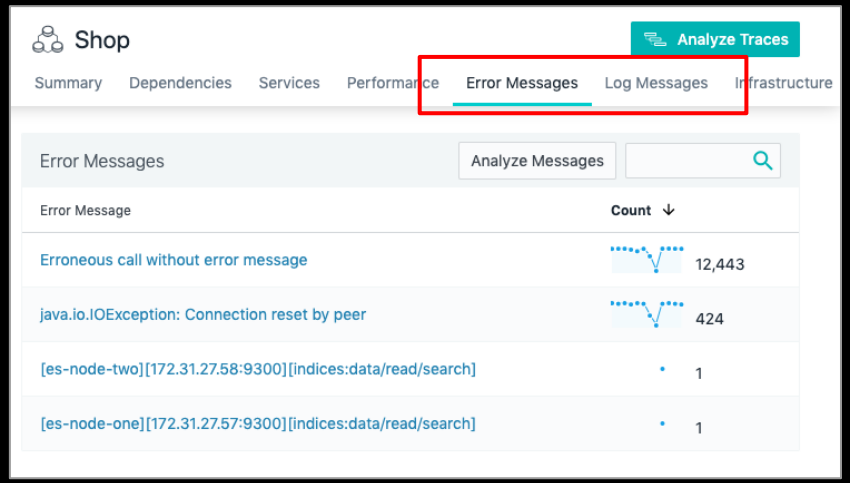




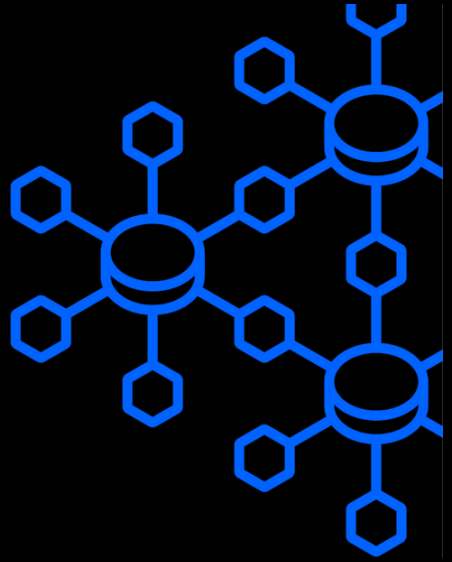
- Application Perspectives
  - Latency
  - Traffic
  - Errors
  - Saturation

Error and Log Messages

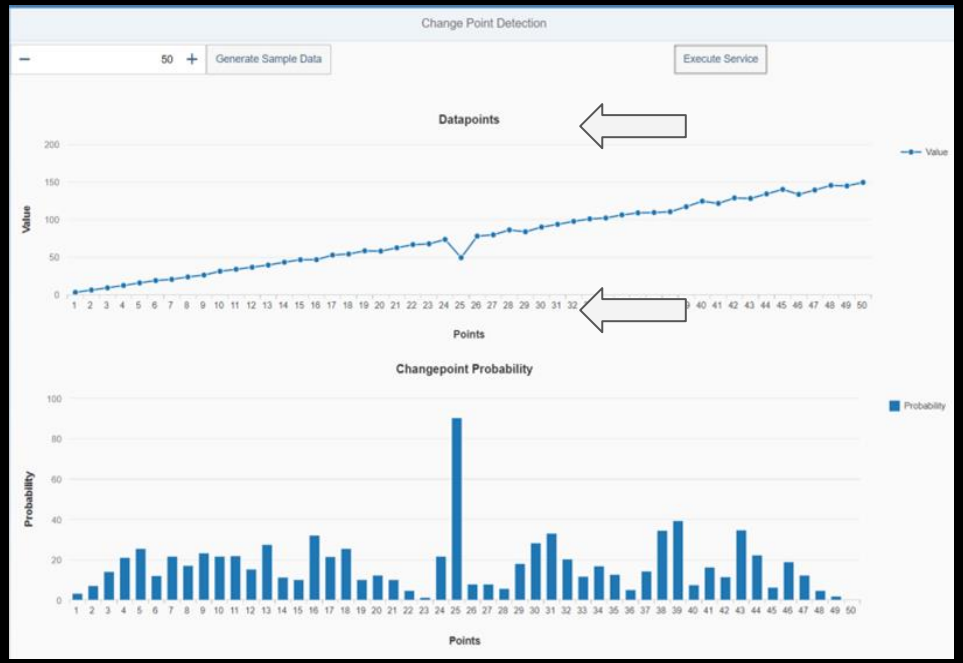
- Error messages
  - Service errors that happen during code execution
- Log Messages
  - Collected from a log message with severity WARN or higher



Intelligent Action



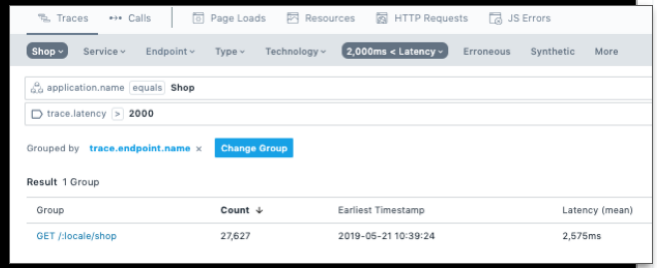
Machine Learning



- Signals Instana trains on
  - Call Rate (sudden drops)
  - Error Rate (sudden increase)
  - Latency (sudden increase)
- Signals tracked from a variety of sources
  - Traces
    - Endpoint, services, app perspectives
  - Metrics

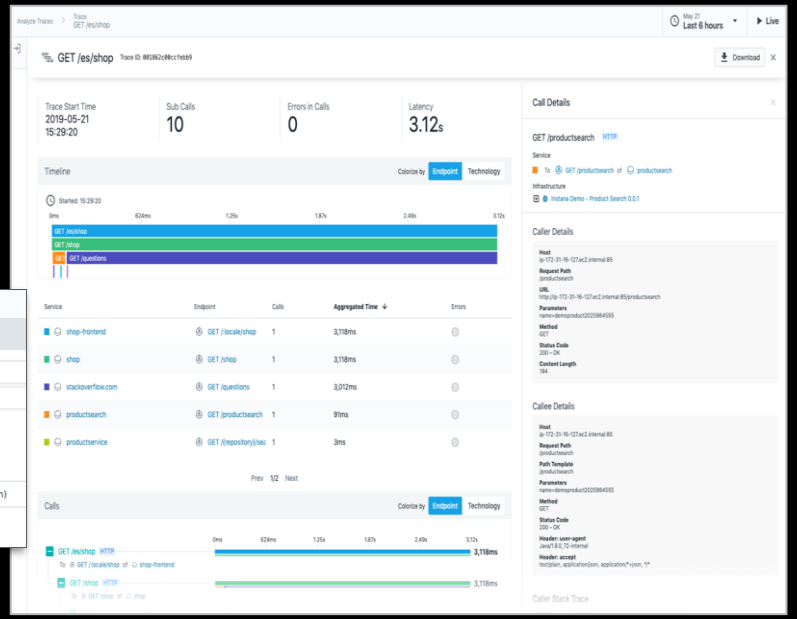
Unbounded Analytics

- Unbounded Analytics focuses on
  - Distributed Traces
  - Logs
  - End User Monitoring



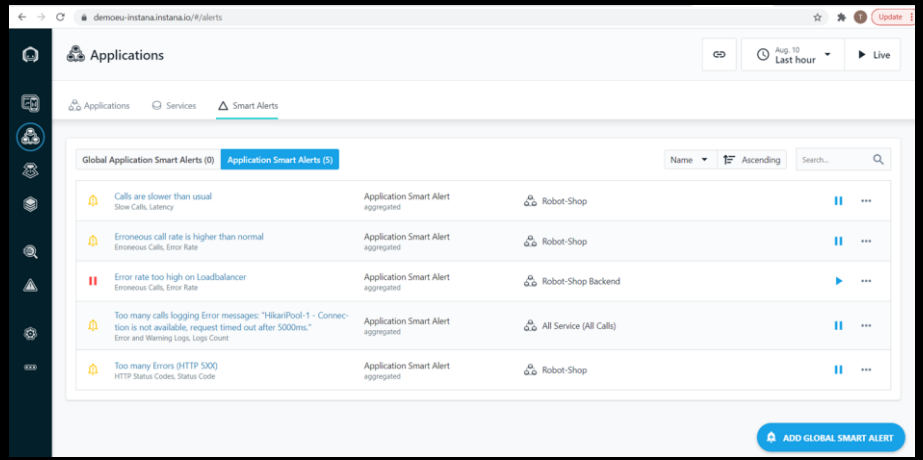
Distributed Trace Analytics

Finds EVERY slow request



Smart Alerts

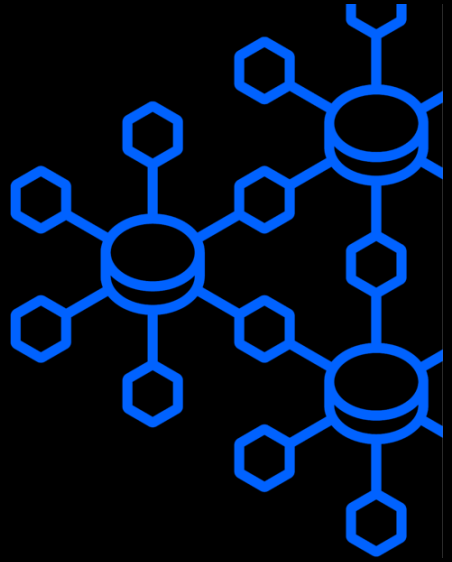
- Use Case Based Alerting
  - Alert suggestions and recommendations
    - Performance, Availability, Errors, Bugs
  - Automated and manageable alerts
    - Customizable Scenarios, Real Time visualization, Seasonality
  - Arbitrary filtering
    - Scope limitations, Traffic narrowing



Instana

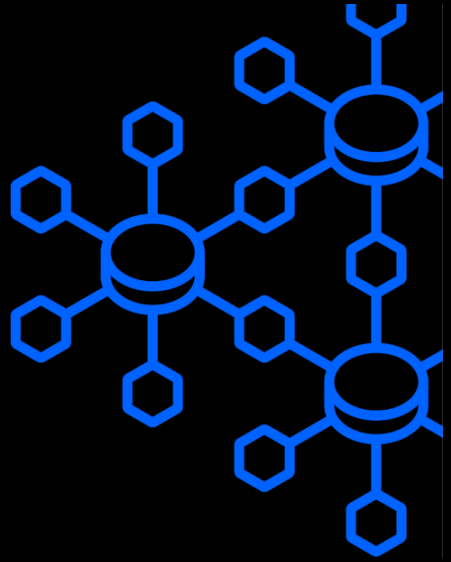
is

Enterprise Observability



Instana Values for Key Stakeholders

- Developers
  - Test new code functionality before committing
- DevOps
  - Enable smooth CI/CD pipeline integration
- SREs
  - Ensure pre and in-production reliability and availability
- Ops
  - Continuously monitor and respond to potential problems and alerts generated by Machine Learning and AIOps



Application Modernization in the Cloud Age

INSTANA
an IBM Company

Tom Fisher

Thank You!