Monitoring to Observability
Empowering DevOps & Digital Business

DevOps Talks Conference 2022

splunk® turn data into doing®
Koray Harman
Observability Strategist
Foodie, Techie, Gamer, Runner, Jazz Enthusiast
Agenda

- DevOps KPIs
- What is Observability
- Pillars of Observability
- Key to digital business
- Hitting the ground running
Waterfall vs Agile Development

Waterfall
- Requirements
- Design
- Development
- Testing
- Deployment

Agile
- Req's
- Launch
- Deploy
- Test
- Develop
- Design
- Req's
- Launch
Devops & Observability

- DevOps has become the standard
- Tensions and difficulties hamper ability to make the most of continuous delivery
- Major factor is prioritisation of delivery speed over visibility
- Closing the gap
  - Just as much effort in measuring software at every stage
  - Early visibility and prioritisation of application performance and user experience
Key DevOps KPIs

- Deployment frequency
- Failed deployment frequency
- Feature/build release rate
- Mean time to discovery (MTTD)
- Mean time to recovery (MTTR)
- Mean lead time (MLT)
- Up-time
- Defect escape rates
- Application performance
What is Observability?
Fun Fact

Numeronym

‘O11y’
Key Observability Objectives

**Customer Experience**
- Reduce risk of lost revenue
- Improve & maintain response times
- Provide high service availability
- Anticipate and prevent issues
- Fast issue resolution time before customers experience degradation
- Real-time visibility into the End User Experience
- Real-time visibility into Customer Journeys

**Optimise Costs**
- Baseline and track costs of cloud services
- Compute optimisation recommendations to reduce cloud spending
- Visibility into the costs of APIs & Services
- Visibility into the costs of service for each customer or business unit
- Reduce Total Cost of Ownership

**Accelerate Development**
- On-time projects, cloud deployments, modernisation
- Accelerate speed of cloud migration/modernisation initiatives
- Validate success of deployments to cloud
- Visibility into business AND technical KPIs and performance
- Attract & retain top talent

**Reduce Unplanned Work**
- Mitigate outages and service impacts
- Reduce MTTR
- Identify and resolve issues before being deployed into production
- Pinpoint root causes to minimise the number of people and teams involved in issue resolution
- Incorporate performance into deployment pipelines to prevent surprises
Every company is on a cloud journey
To increase velocity, agility and responsiveness

Retain & Optimize

Lift & Shift

Re-Factor

Re-Architect / Cloud-Native

Tightly Coupled Apps, Slow Deployment Cycles

Primarily using Cloud IaaS

More Modular, but Dependent App Components

Loosely Coupled Microservices, and Serverless Functions

Cloud First Architecture

Cloud Managed e.g. RDS, DynamoDB, SaaS
Your World Has Never Been More Complex

Retain & Optimize

Lift & Shift

Re-Factor

Re-Architect / Cloud-Native

Tightly Coupled Apps, Slow Deployment Cycles

Primarily using Cloud IaaS

More Modular, but Dependent App Components

Loosely Coupled Microservices, and Serverless Functions

Baggage Java App

Miles App

Flight Status Service

Checkout Microservice

Boarding Pass Microservice

Airport kiosk

Online booking

Mobile app

Databases, App Servers, VMs in private datacenter

ELB + RDS + MongoDB + AWS EKS

Serverless functions on Azure and GCP

Your World Has Never Been More Complex

Tightly Coupled Apps, Slow Deployment Cycles

Primarily using Cloud IaaS

More Modular, but Dependent App Components

Loosely Coupled Microservices, and Serverless Functions

Baggage Java App

Miles App

Flight Status Service

Checkout Microservice

Boarding Pass Microservice

Databases, App Servers, VMs in private datacenter

ELB + RDS + MongoDB + AWS EKS

Serverless functions on Azure and GCP
Complexity Can Kill Innovation
What’s Required for Observability

- Full-Stack and End-to-End
- Full-Fidelity
- Real-Time
- Analytics-Powered
- Metrics
- Traces
- Logs
- OpenTelemetry
- Scalable
- Enterprise-Grade
Observability
The Three Pillars

WHAT’S HAPPENING?

METRICS
Detect
Observability
The Three Pillars

WHAT’S HAPPENING?

WHERE IS IT HAPPENING?

METRICS
Detect

TRACES
Troubleshoot
Observability

The Three Pillars

WHAT’S HAPPENING?

WHERE IS IT HAPPENING?

WHY IS IT HAPPENING?

METRICS
Detect

TRACES
Troubleshoot

EVENTS / LOGS
Pinpoint
Observability

The Three Pillars

WHAT’S HAPPENING?
WHERE IS IT HAPPENING?
WHY IS IT HAPPENING?

METRICS
Detect

TRACES
Troubleshoot

EVENTS / LOGS
Pinpoint
The Five Tenets of Observability

- **Full Stack & Full Fidelity**: Capture 100% of telemetry data across all layers of your stack.
- **Real Time**: See impact of changes as they happen and alerts in seconds, not minutes.
- **Analytics-Powered**: KPI’s, business insights and analytics for directed troubleshooting and improving service performance.
- **Enterprise Grade + Scale**: Massively scalable, teams and permissions, fully programmable, monitoring-as-code, usage reporting and control, built-in security.
- **Open Standards**: Use open source projects to prevent vendor lock-in.
Leveraging the wider community

Open Standards for Data Ingestion

- Combine **Distributed Tracing**, **Metrics**, and **Logging** into a single set of system components and language-specific libraries

- Several community initiatives towards the same goal, combine and merge efforts
What is OpenTelemetry?

OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software.

OpenTelemetry provides a single set of APIs, libraries, agents, and collector services to capture distributed traces and metrics from your application. You can analyze them using Prometheus, Jaeger, and other observability tools.
# OpenTelemetry Components

<table>
<thead>
<tr>
<th>Specification</th>
<th>Collector</th>
<th>Client Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>API: Baggage, tracing, metrics</td>
<td>Receive, process, and export data</td>
<td>Application instrumentation</td>
</tr>
<tr>
<td>SDK: Tracing, metrics, resource, configuration</td>
<td>Default way to collect from instrumented apps</td>
<td>Support for traces, metrics and logs</td>
</tr>
<tr>
<td>Data: Semantic conventions, protocol</td>
<td>Can be deployed as an agent or service</td>
<td>Mobile &amp; Browser Instrumentation</td>
</tr>
</tbody>
</table>
Performance matters
Web and mobile performance impact business outcome

- 10% improvement in LCP
- 100ms improvement raised revenue 1%
- 8% increased sales
- 24% less user abandonment

- BBC
- WALMART
- VODAFONE
- GOOGLE
Observability Moves the Needle
More productive developers and happier customers

- **Release Quality and Velocity**: 8X
  - Higher quality and faster code releases for new launches and updates

- **Customer Experience**: 100X
  - Fewer missed anomalies and end user-impacting incidents as well as ability to effectively optimise experience

- **System Availability**: 80%
  - Faster identification, acknowledgement and resolution of defects

- **Developer Efficiency**: 70%
  - Reduction in developer disruptions from fewer incidents and less time spent troubleshooting defects
Unlocking Business Insights
Maturing Observability Use-Cases

- Correlating technical investments to business KPIs
  - Revenue growth per sprint/release/feature
  - Production efficiency (i.e. widget production) per sprint/release/feature
  - Deeper understanding of cost from performance impacts and outages
  - Cost of service (APIs, serverless functions)
  - Pivoting by customer, region, group, SLA tier, any tag to support your use-case
  - Correlating User Experience (UX) and business KPIs (sales, revenue, conversion, etc.)

- Validating business assumptions
  - Is our understanding of UX translating to KPIs?
  - Criticality of services and their impacts to business KPIs
  - Investment vs return (Engineering to KPI linkage)
Hitting the ground running with Observability

- Start Early & Mature Together
- Embed Observability into all stages of the DevOps loop
- Use Open Standards for data ingestion (OpenTelemetry)
Thank You