SKILUP DAYS by: DevOps Institute

Observability in Modern Applications: Simplifying Complex Systems

Devon Lawler - Director of Sales Engineering @epsagon

epsagon

About Me

- Director of Sales Engineering
- @devonlawler1
- <- ~15 minutes after my wedding!

epsagon

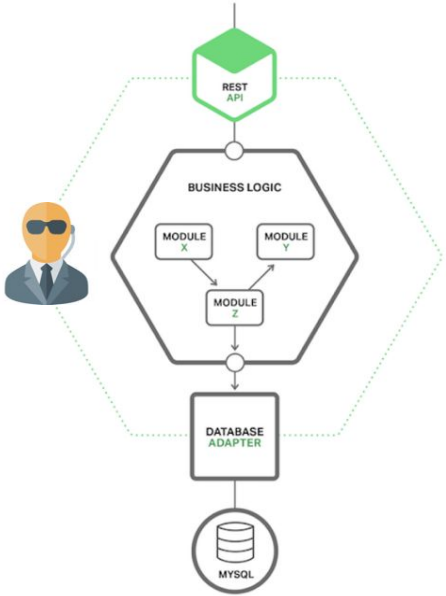
What We'll Discuss Today

- Old-School Monitoring Approaches
- Troubleshooting Pitfalls
- Observability in Distributed Environments
- In-House vs. Managed Solutions

epsagon

Old-school Monitoring

- Distributed logs
- Collects only host data
- Collects only metrics



epsagon

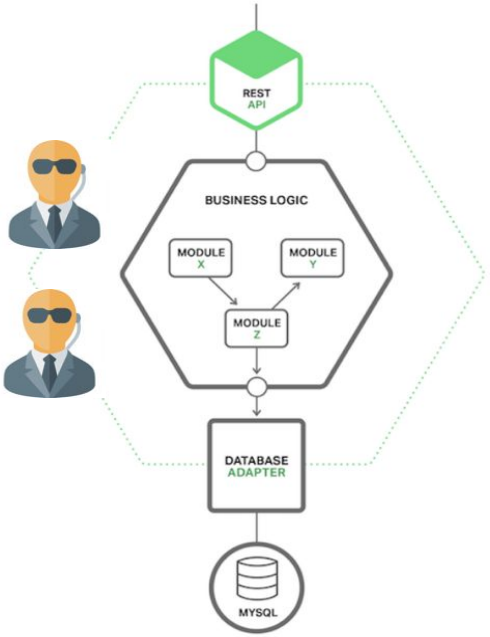
Troubleshooting

We need more debug data -> logs

epsagon

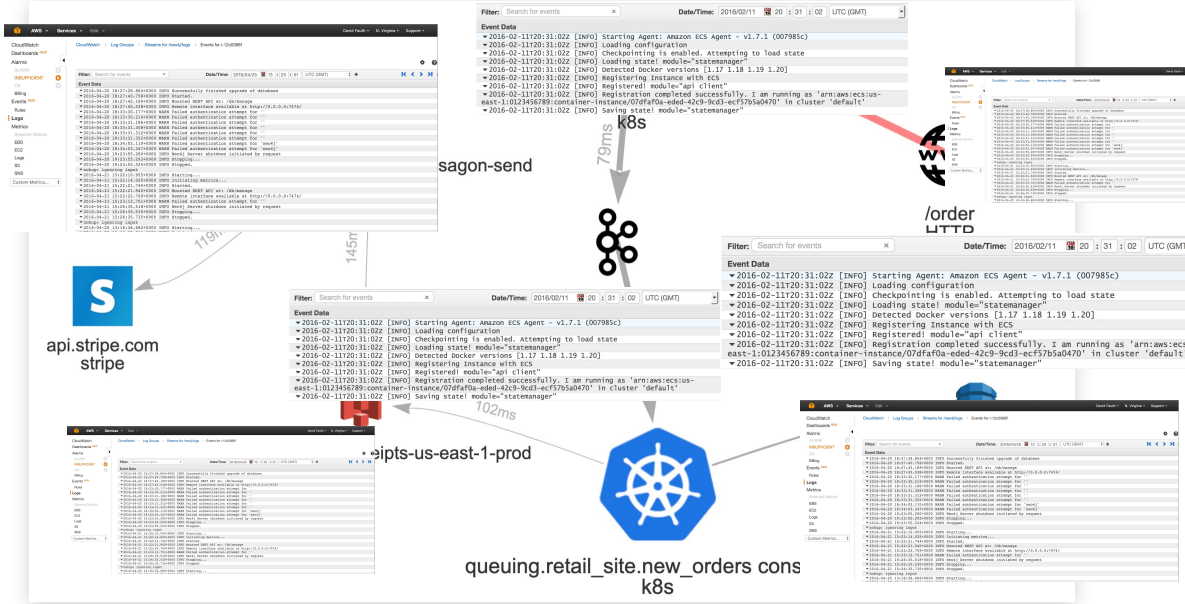
Old-school Logging

- YaA (Yet another Agent)
- Expensive
- Collects only logged data



epsagon

Distributed Logs Don’t Scale



epsagon

Challenges for Engineering and DevOps

Development	Monitoring	Troubleshooting
"I'm not sure what's currently running in production. How can I build new services?"	"Is my app working properly?"	"Are basic logs and metrics the right tool for highly distributed applications?"

epsagon

The Three Pillars of Observability

Combining metrics, logs, and traces for observability is the **only** way to understand complex environments

Metrics tell us the "what"

Logs tell us the "why"

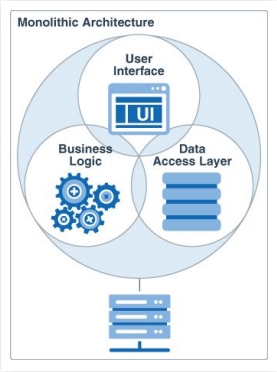
Traces tell us the "where"

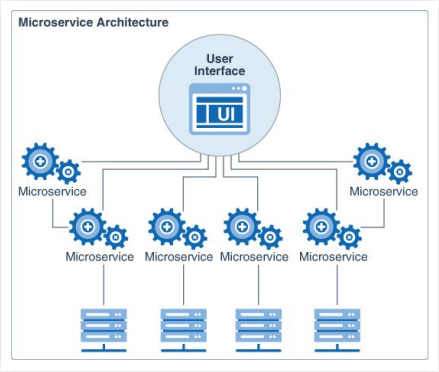
epsagon

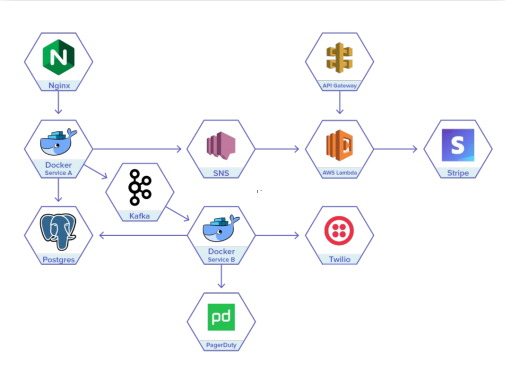
Something is Still Missing

- Manual Correlation
- Multiple Sources
- Single pane of glass

The Rise of Microservices in the Cloud





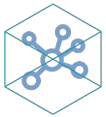


Host-based Monolithic → Host-based Distributed → Abstracted-host Highly Distributed

Extremely hard to monitor and troubleshoot

epsagon

Traditional Monitoring Solutions are Limited



Logs & Metrics are NOT Enough

- Traces are needed on top of metrics and logs
- Distributed tracing is crucial in order to find the root-cause efficiently

Bytecode becomes a bottleneck

- Bytecode provides limited value in distributed applications
- Bytecode comes with significant overhead in microservice environments

Manual Monitoring Kills Agility

- Getting distributed traces require manual instrumentation
- Metrics, logs and traces are correlated manually

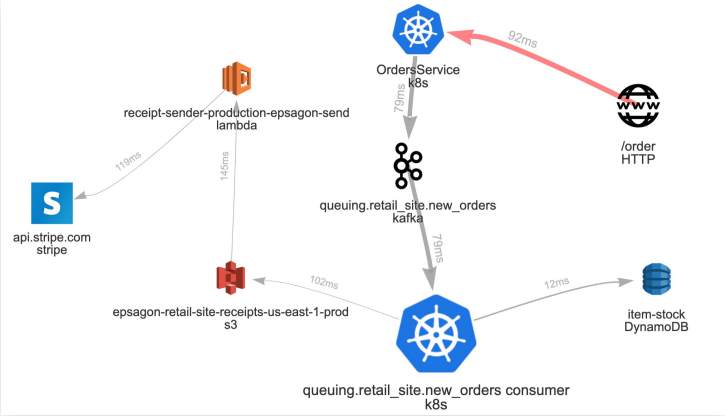
epsagon

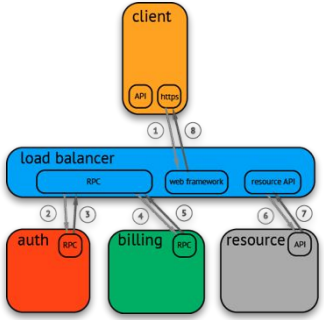
Why Distributed Tracing Is Critical Today

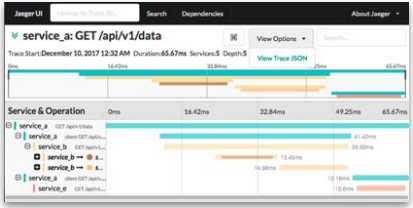
The Only Way to Understand Cloud-Native Workloads

Due to high complexity and the need for manual instrumentation, distributed tracing remained an approach viable only for very **tech savvy** companies

> A trace tells the story of a transaction or workflow as it propagates through a distributed system







epsagon

Open-source Tools

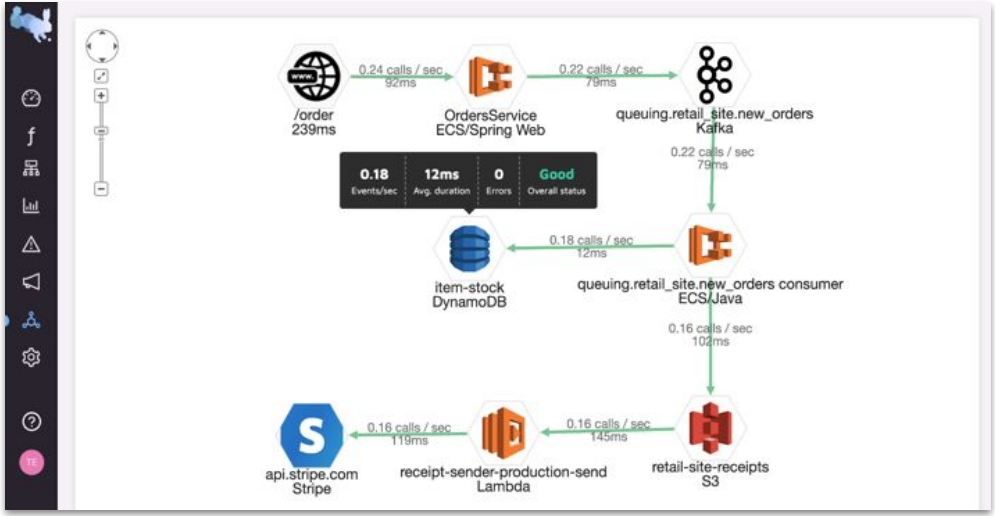
Generating Traces

Ingestion & Client

Best Practices for Observability

- **Automated** setup and zero maintenance
- **Supports any environment**

  (K8s, cloud, FaaS)
- Connects **every request** in a transaction
- **Searches and analyzes** your data
- **Helps** to quickly pinpoint problems
- **Correlation**



epsagon

Observability Benefits

- Reduction in Error Rates
- Reduction in Troubleshooting Time
- Faster Shipment of Features
- Improved DevOps & Engineering Efficiency

epsagon

The Journey to Observability

- Identify your business goals and architecture model
- Determine your approach: DIY or managed
- Trial observability solutions
- Make sure the new service integrates to your ecosystem

epsagon

Summary

- Modern applications require more than just monitoring
- Distributed tracing is a crucial component in such environments
- Automation and Unification for efficiency and ease
- Stop implementing your own solutions unless needed

epsagon

Special Offer for SkilUp Day: Observability

Visit https://epsagon.com/skilupday/

Start a free trial, send your first trace & we'll send you our Cloud Observability Drone

1-in-10 will also receive a pair of Bose Headphones!

epsagon

Thank you!

Meet me in the Network Chat Lounge for Questions.

epsagon