ARQUITETURA DE SW

EAD - Logging / Elastic Stack

Análise e Desenvolvimento de Sistemas

Prof. Ms. Renato Leite
renato@sectras.edu.br

OBJETIVOS

1. Contexto e desafios da operação de microservices

2. Centralização de logs e métricas

3. Monitoramento distribuído e Circuit Breaker

4. Automação e malha de observabilidade

5. Exercícios

REVISANDO DEVOPS

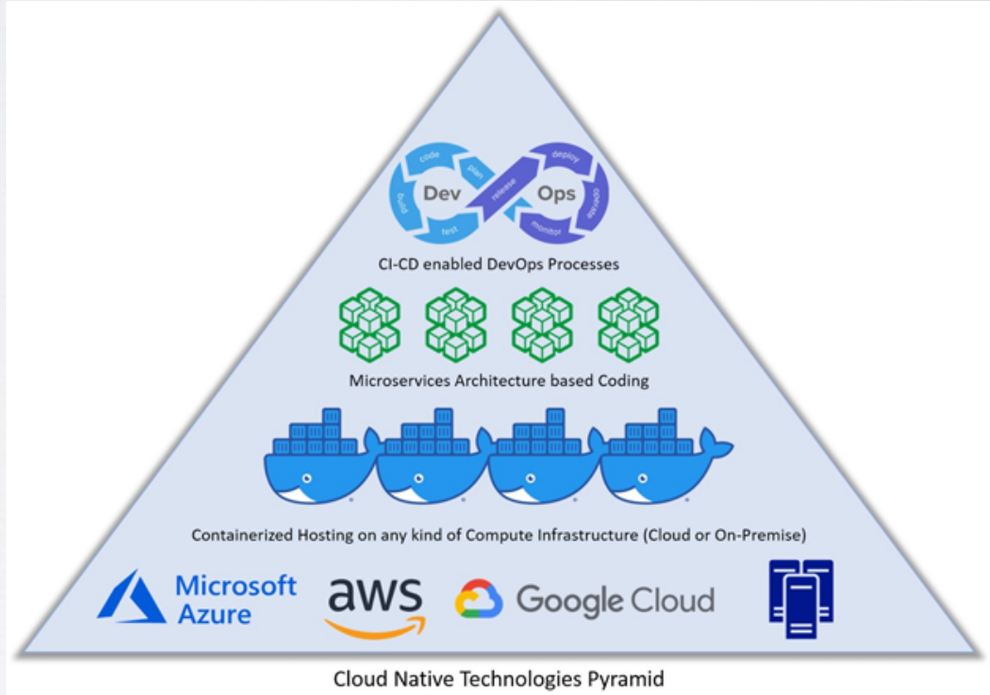
1. Com o avanço das arquiteturas de microservices, surgem novos desafios de monitoramento e controle.

2. Cada serviço é independente, com logs e métricas próprios.

3. O DevOps passa a incluir **observabilidade**, **resiliência** e **automação operacional**.

4. A confiabilidade deixa de ser centralizada e passa a emergir da cooperação entre serviços.

5. Em DevOps, operação e desenvolvimento formam um ciclo contínuo de aprendizado.



Cloud Native Technologies Pyramid

CONTEXTO DE LOGGING

1. Erro (Error): ação humana incorreta durante o desenvolvimento ou operação.

2. Defeito (Defect): materialização do erro no código, configuração ou pipeline.

3. Falha (Failure): comportamento incorreto em execução, percebido pelo usuário.

4. Logging: atua como ponte entre essas etapas, registrando o que ocorreu, onde e sob quais condições.

5. Com logs estruturados e centralizados, é possível rastrear a origem, a causa e o impacto de cada falha no ciclo de vida do software.

LOGGING - FASES

1. Depuração (1970–1990): o log nasceu como ferramenta técnica para identificar **erros** após a execução.

2. Operação (2000–2010): passou a apoiar equipes de **infraestrutura** na detecção e **diagnóstico de falhas**.

3. **Observabilidade** (2010–2020): tornou-se pilar do DevOps, integrando logs, métricas e traces em **tempo real**.

4. **Compliance** (2020–presente): virou evidência de conformidade e rastreabilidade exigida por **regulações** e

ELASTIC STACK

1. Ambientes distribuídos exigem consolidação de logs e métricas.

2. Ferramentas como ELK Stack (Elasticsearch, Logstash e Kibana) e Prometheus + Grafana centralizam e analisam dados.

3. Essas soluções permitem detectar anomalias e criar alertas inteligentes.

LOGSTASH

- Ferramenta de envio e transformação de dados.
- Age como um pipeline que coleta logs de diversas fontes, aplica filtros e envia os dados processados ao ElasticSearch ou outros destinos.
- Características:
    - Suporte a centenas de inputs e outputs (arquivos, Kafka, Beats, S3, etc.).
    - Filtros potentes: grok, json, mutate, geoip, date.
    - Possui arquitetura modular e extensível por plugins.
    - Permite enriquecimento dos dados antes do armazenamento (ex.: origem, host, IP, serviço).
- Aplicações:
    - Centralização e padronização de logs de containers e microservices.
    - Envio simultâneo de dados para ElasticSearch, Kafka e Data Lakes.
    - Base para pipelines observáveis e auditáveis.

ELASTICSEARCH

- Plataforma de busca, indexação e análise em tempo real.
- É o núcleo da pilha ELK, responsável por armazenar e permitir consultas rápidas sobre logs, métricas e eventos.
- Características:
    - Armazena dados em índices distribuídos (clusters horizontais).
    - Permite consultas complexas (full-text search) e agregações numéricas.
    - Altamente escalável e resiliente.
    - Suporta replicação automática, sharding e compressão de dados.
- Aplicações:
    - Observabilidade e auditoria de microservices.
    - Facilita rastreamento de falhas, análise de desempenho e correlação de eventos.
    - Pode ser integrado a sistemas de alerta (Watcher, Grafana, Prometheus).

KIBANA

- Ferramenta de visualização, análise e interação com os dados armazenados no ElasticSearch.
- Oferece dashboards dinâmicos e consultas visuais sobre logs, métricas e eventos.
- Características:
    - Interface web integrada à Elastic Stack.
    - Criação de dashboards interativos e filtros customizados.
    - Suporte a gráficos, mapas, timelines e relatórios.
    - Permite alertas e monitoramento em tempo real.
- Aplicações em DevOps:
    - Dashboards de latência, erros e disponibilidade de microservices.
    - Análise de logs e rastreamento de incidentes (por tempo, serviço ou severidade).
    - Integração com Elastic APM para visualização de traces e transactions.

KIBANA


Logstash Search
2 days ago to 3 minutes ago
QUERY
bytes:[0 TO 4000000] AND @tags:success
FILTERING
EVENTS OVER TIME
View | Zoom Out | html (ext) (868) php (ext) (659) png (ext) (317) gif (ext) (229) css (ext) (121) count per 30m | (2214 hits)
60
50
40
30
20
10
0
00:00 02-24
04:00 02-24
08:00 02-24
12:00 02-24
16:00 02-24
20:00 02-24
00:00 02-25
04:00 02-25
08:00 02-25
12:00 02-25
16:00 02-25
20:00 02-25
MAP
REVENUE
22%
VOLUME
9%
TOP DESTINATIONS
Other values 42%
CN 19%
IN 16%
TOP SOURCES
Other values 52%
IN 19%
US 10%
GEO PAIRS
100
80
60
40
20
0
ALL EVENTS
0 to 100 of 500 available for paging
Fields
All (21) / Current (20)
Type to filter...
@message
@tags
@tags
geo.srcdst
extension
clientip
bytes
id
phpmemory
response
success,security
MY:VN
html
167.12.22.189
8540
1066
200
success,info
IT:MM
png
164.87.170.73
2045
1903
200
success,info
AR:ES
html
222.23.102.238
1801
1133
200
success,info
IN:DZ
html
138.226.66.81
7029
1801
200

CASE ELK BELL CANADA

- A Bell Canada enfrentava limitações em seu sistema de monitoramento e segurança (SIEM legado), incapaz de lidar com o volume crescente de logs de rede, servidores, containers e aplicações. O excesso de falsos positivos e a dificuldade de correlacionar eventos prejudicavam a eficiência operacional.
- Solução implementada: Substituição do sistema anterior por uma arquitetura baseada em Elastic Stack (ELK).
  - Filebeat e Winlogbeat para coleta de logs de múltiplas fontes.
  - Logstash para normalização, filtragem e enriquecimento dos dados.
  - Elasticsearch para armazenamento e indexação em cluster escalável.
  - Kibana para visualização e análise em tempo real.
  - Integração com alertas e controle de acesso via RBAC.
- Resultados alcançados:
  - Redução significativa de falsos positivos em alertas de segurança.
  - Escalabilidade horizontal e alta disponibilidade na ingestão de logs.
  - Agilidade para adicionar novas fontes de log sem reconfiguração complexa.
  - Melhoria na visibilidade operacional e na resposta a incidentes.

https://www.elastic.co/elasticon/tour/2018/toronto/security-events-logging-bell-canada

LOGGING DATA LAKES

- À medida que sistemas em microservices evoluem, o volume de logs e métricas cresce. Apenas ELK para fornecer dados em tempo real já não é suficiente;

- É preciso prever! E qual o grande insumo de qualquer previsão?

- Armazenar e processar grandes volumes históricos de dados operacionais se torna necessário;

- Tornando o ambiente capaz de aprender com o passado para antecipar falhas futuras — é aí que entram os data lakes.

LOGGING DATA LAKES

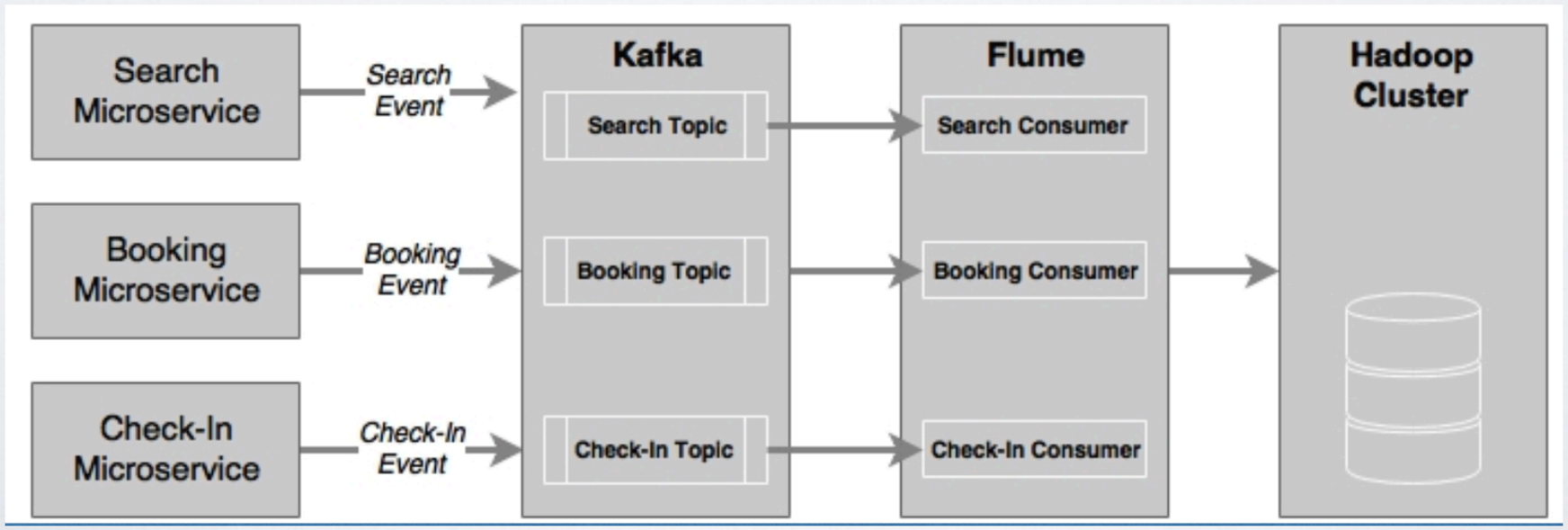
- À medida que sistemas em microservices evoluem, o volume de logs e métricas cresce. Apenas ELK para fornecer dados em tempo real já não é suficiente;

- É preciso **prever**! E qual o grande insumo de qualquer previsão?

- **Armazenar** e processar grandes volumes **históricos** de dados operacionais se torna necessário;

- Tornando o ambiente capaz de aprender com o passado para **antecipar falhas** futuras — é aí que entram os data lakes.

- Hadoop é uma das ferramentas usadas para gerenciar o DataLake



EXERCÍCIO

1. Melhore a solução de logging implementada na última aula para incluir o LogStash (ou ferramenta de transformação de dados sua preferência) + Kibana (visualização)

2. Ponto Extra (1,0): implemente um DataLake usando Hadoop.

BIBLIOGRAFIA

1. GALLOTTI, G. M. A. Arquitetura de Software. 1ª Ed. São Paulo: Pearson, 2016. [acervo digital]

2. JUNIOR, A. K. Computação em Nuvem. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]

3. ZANETTI, M. Arquitetura de TI e modelos de negócios. 1ª Ed. Curitiba: Contentus, 2020. [acervo digital]

4. Baseado nos materiais de aula de Vinícius Garcia - CIn UFPE

DÚVIDAS?