

Crie métricas de observabilidade para o processo ETL que você desenvolveu anteriormente (não é necessário implementação):

1. Monitore o tempo que leva para os dados serem extraídos, transformados e carregados.

Métricas:

- Tempo de Extração
- Tempo de Transformação
- Tempo de Carregamento

Para isso, podemos utilizar o serviço Amazon Cloudwatch, através dos logs, é possível acompanhar toda a execução de cada etapa do ETL e monitorar o seu tempo. Além disso, é possível criarmos métricas personalizadas para monitorar o tempo de execução de cada etapa do processo.

Como fazer isso?

1. Use logs do CloudWatch para registrar eventos e erros durante o processo.
2. Configure métricas personalizadas exemplo:
  - a. tempo\_extracao\_segundos.
  - b. tempo\_transformacao\_segundos.
  - c. tempo\_carregamento\_segundos.
3. Configure dashboards no CloudWatch para visualizar as métricas em tempo real.
4. Envie alertas caso a taxa de alguma dessas métricas esteja abaixo do que foi definido no SLA.

Exemplo de Alertas:

1. Alerte se o tempo de execução de qualquer etapa exceder um limite pré-definido pelo SLA.
2. Alerte se a taxa de falhas for maior que 0.

2. Implemente alertas para qualquer falha ou anomalia durante o processo ETL.

Para detectar casos de falha ou anomalia durante o processo de ETL, podemos utilizar os pilares de Data Quality para criar métricas de observabilidade personalizadas para os nossos dados e utilizar as métricas padrão para acompanhar a execução dos serviços no CloudWatch:

<b>Categoria</b>	<b>Descrição</b>
<b>Temporalidade</b>	Atraso na chegada dos dados (latência)
<b>Integridade</b>	Quantidade de registros corrompidos ou falhas no pipeline
<b>Unicidade</b>	Duplicatas de IDs únicos (ex.: transaction_id)
<b>Validade</b>	Registros com valores inválidos (ex.: datas no futuro, valores negativos)
<b>Compleitude</b>	Percentual de valores nulos ou faltantes

### **Fluxo Completo de Alerta para os ETLs:**

#### **1. Coleta de Métricas:**

O **CloudWatch** coleta métricas do processo ETL (tempo de execução, falhas, volume de dados, etc.).

#### **2. Detecção de Anomalias:**

Os alarmes do **CloudWatch** monitoram as métricas e são acionados quando os limites são ultrapassados.

#### **3. Notificação:**

O **SNS** envia notificações para a equipe (e-mail, SMS) quando um alarme é acionado.

#### **4. Ação Automatizada:**

O AWS Lambda executa ações automatizadas em resposta a alarmes (por exemplo, reiniciar um job no Glue de ETL).

#### **5. Visualização:**

A equipe monitora o processo ETL em dashboards do CloudWatch.

3. Descreva como você rastrearia um problema no pipeline, desde o alerta até a fonte do problema.

### **1. Recebimento do Alerta**

Nessa etapa é importante identificar através do alerta qual métrica falhou e se é um problema recorrente. Além disso, é importante identificar qual serviço ocorreu a falha, o erro relatado e em qual etapa do pipeline ele se encontra.

### **2. Analisar logs da Falha**

Nessa etapa, podemos utilizar o CloudWatch para acessarmos os logs do serviço e entender melhor de que se trata.

### **3. Identificar e investigar o problema e Reprodução**

Aqui, conseguimos identificar o problema no serviço e qual erro ocorreu e, se necessário, tentar reproduzi-lo no ambiente de testes. Aqui, também podemos identificar se há registros passados desse erro e ele voltou a ocorrer e os possíveis impactos causados no pipeline.

Caso seja um problema que não possa ser corrigido a tempo e irá gerar indisponibilidade dos dados, é necessário preparar um comunicado às equipes e times que o consomem para ser enviado por e-mail ou pelos canais de comunicação utilizados pelas equipes.

### **4. Correção do Problema**

Com base nas análises da etapa 3, priorizamos a correção desse problema na sprint como **"Fura Fila"**, e alocamos capacity para a correção do erro. Importante que sejam feitos testes e que essa correção passe por todo fluxo da esteira dos ambientes.

### **5. Documentação, Prevenção e Monitoramento**

Nessa etapa, documentamos o erro ocorrido e como foi corrigido para auxiliar em futuras investigações de erros que possam surgir, e, executamos o pipeline corrigido. Seguimos monitorando para ver se o fluxo voltou a operar normalmente.