

Primeiro Trabalho de Distribuição e Concorrência de 2025/2

Instruções gerais:

- Trabalho individual ou em grupo de até dois componentes.
- Data da entrega: ver EaD.
- Data para apresentação: ver EaD.

Tema:

Balanceamento de Carga e Failover com Kafka em Clusters Docker.

Mini-mundo: Sistema de Monitoramento de Sensores em uma Fábrica Inteligente.

Descrição:

Uma fábrica inteligente possui diversas máquinas espalhadas em diferentes setores (linha de produção, refrigeração, empacotamento, etc.). Cada máquina é equipada com sensores que enviam dados (temperatura, vibração, consumo de energia, etc.) continuamente para um sistema central que processa esses dados em tempo real para detectar anomalias, falhas ou padrões de uso.

A equipe de engenharia precisa que o sistema:

- Seja tolerante a falhas (ou seja, não pare se uma parte cair),
- Escale com facilidade conforme o número de sensores aumenta,
- E balanceie a carga de forma eficiente entre processadores de dados.

Componentes do Mini-Mundo

1. Sensores (Produtores)
 - Simulados por containers Docker que geram dados periódicos (ex: JSON com temperatura e vibração).
 - Cada sensor envia mensagens para um tópico Kafka chamado `dados-sensores`.
2. Cluster Kafka (Brokers)
 - 2 ou mais instâncias de Kafka rodando em containers diferentes.
 - Cada tópico é dividido em várias partições, com replicação configurada.
3. Consumidores (Processadores de dados)
 - Aplicações Java (ou Python) que consomem mensagens do tópico `dados-sensores`.
 - Elas processam os dados e detectam, por exemplo, quando a temperatura ultrapassa limites.
 - Os consumidores pertencem ao mesmo grupo de consumo, compartilhando a carga (um lê partição A, outro a B...).
4. Banco de dados ou logger
 - Os dados processados (ou alertas) são salvos ou registrados para análise posterior.

Objetivos do Trabalho

1. Criar um cluster Kafka com múltiplos brokers e configurar tópicos com partições e replicação.
2. Simular sensores como produtores Kafka em containers Docker distintos.
3. Implementar múltiplos consumidores Kafka com balanceamento de carga automático.
4. Simular falhas:
 - Parar um broker Kafka e mostrar que o sistema continua funcionando.
 - Parar um consumidor e mostrar que outro assume sua partição (rebalanceio).
5. Demonstrar visualmente ou via logs o comportamento do sistema sob carga e falhas.

Entrega:

O trabalho deverá ser entregue via EaD, incluindo um relatório do que foi implementado contendo o que funcionou e o que não funcionou. Deve ter também documentação de como instalar a aplicação assim como instruções de como operar.

O trabalho deverá ser apresentado pelo grupo em data a ser marcada. Todos os componentes do grupo devem ser capazes de responder sobre qualquer parte do trabalho. Embora o trabalho seja em grupo, as notas serão individuais.

Entregáveis:

- Relatório
 - Documentação de instalação e uso
 - Explicação da arquitetura
 - Testes de falha
 - Exibição dos resultados
- Makefile
- Código-fonte dos produtores e consumidores
- Arquivos pom.xml
- Arquivo docker-compose.yml com todos os serviços
- Scripts de simulação de falhas
- Logs de execução mostrando rebalço

Dicas (que você vai seguir porque elas serão pontuadas)

1. Comente o seu código usando JavaDoc ou DocString
2. Não use constantes *hard-coded*, crie um arquivo `application.properties` ou variáveis de ambiente no arquivo `docker-compose.yml`.

Na dúvida, não invente, pergunte!