Tutorial Publish-Subscribe com Kafka

### **Prompt inicial para chatgpt (opcional)**

| Me ajude a criar um exemplo simples de aplicação que usa Kafka para publish-subscribe. A aplicação diz respeito ao sistema de monitoramentos de voos de um aeroporto. O publisher será o AirportDatabase, e o subscribe serão os Tótens de Voos (Flight Totem), que podem ser para voo de chegada, partida, ou ambos.  1) Me ajude a criar um deployment do Kafka usando docker  2) Me ajude a configurar dois tópicos: voos de chegada, voos de partida  3) Me ajude a criar um publisher em Golang, que publica vôos nos dois tópicos  3.1) Um voo deve ter os seguintes campos: id, companhia, origem, partida, data e hora.  4) Me ajude a criar 3 subscribers em Golang que imprimirão os voos que forem notificados para eles:  4.1) FlightTotemArrival  4.2) FlightTotemDeparture  4.3) FlightTotemAll |
| --- |

### **Use o docker-compose.yaml do diretório “**[**cap2**](https://drive.google.com/drive/folders/1dNGNm8N36SxGPFJy4UBLuq2n83Yt8fJq?usp=sharing)**” do google drive:**

| docker-compose up -d  # quem for usar cloudshell, substituir por imgs da confluent |
| --- |

### **Configurar dois tópicos: voos de chegada e voos de partida**

| docker exec kafka kafka-topics.sh --create --topic arrival-flights --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1  docker exec kafka kafka-topics.sh --create --topic departure-flights --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1 |
| --- |

### **Você precisa de uma biblioteca Kafka para Go. Vamos usar github.com/segmentio/kafka-go.**

| mkdir airport-monitor  cd airport-monitor  go mod init airport-monitor  go get github.com/segmentio/kafka-go |
| --- |

### **Copie os arquivos .go do diretório “cap2/airport-monitor” do Google Drive para seu diretório local (ou nuvem) chamado airport-monitor**

### **Execute os seguintes comandos, em diferentes terminais, na seguinte exata ordem**

| 1. go run totem\_all.go subscriber.go 2. go run totem\_arrival.go subscriber.go 3. go run totem\_departure.go subscriber.go 4. go run publisher.go |
| --- |