
Sistemas Distribuídos e Mobile

Kafka as a Service

Confluent Kafka Plataforma

Kafka as a Service

- Como abordado em encontros anteriores, uma característica das novas abordagens em nuvem (Cloud) é a venda de soluções tecnológicas como um **Serviço**.
- Manter um infra estrutura de Cluster para funcionamento do Kafka, pode ser onerosa e desgastante dado o aplicação e tamanho do projeto (o que o envolveria o tamanho e gerenciamento do cluster).
- Diante desse cenário, soluções de nuvem para Kafka são oferecidas na forma de **X as a Service**, ou seja, **Plataforma como Serviço**.
- Essas soluções permitem que os desenvolvedores foquem mais na solução da aplicação desenvolvida do que na configuração da plataforma Kafka.

Confluent Kafka Plataforma

- O Confluent Kafka é um serviço que ajuda empresas a criar e gerir pipelines de dados e aplicações de streaming em tempo real. Ele é baseado no Apache Kafka, uma plataforma de streaming que coleta dados de streaming e registra o que aconteceu e quando.

(aws:https://docs.aws.amazon.com/pt_brstreams/latest/dev/using-other-services-read-kafka.html)

- A Confluent oferece uma plataforma de streaming de dados completa e nativa da nuvem, disponível em qualquer lugar que você precisar.
- Confluent permite criar uma conta gratuita para aprendizado, facilitando o uso da plataforma Kafka.

Confluent Kafka Plataforma

- Acesse o link:
 - https://www.confluent.io/get-started/?session_ref=direct
- Cadastrando gratuitamente na plataforma
 - Guarde o código promocional **CONFLUENTDEV1** para criar sua conta gratuita sem a necessidade de inserir um cartão de crédito para utilização da plataforma.

Introdução

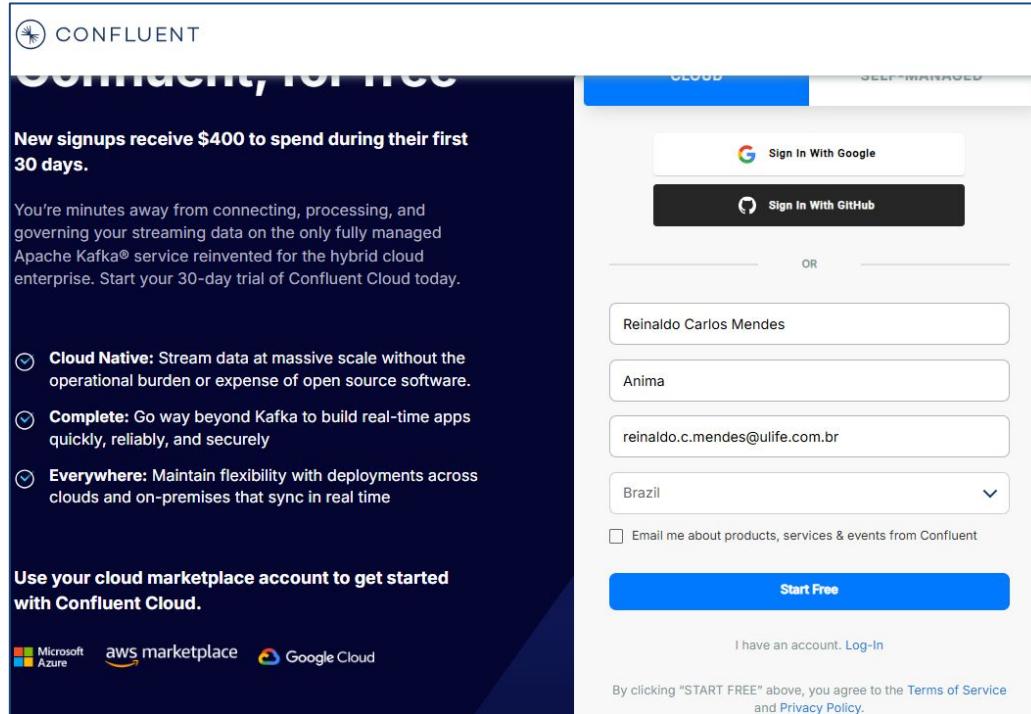
Neste tutorial, você criará aplicativos cliente Python que produzem e consomem mensagens de um cluster Apache Kafka®.

Conforme você aprende a executar seu primeiro aplicativo Kafka, recomendamos usar o Confluent Cloud para que você não precise executar seu próprio cluster Kafka e possa se concentrar no desenvolvimento do cliente. Se você ainda não tem uma conta, certifique-se de [se inscrever](#). Novas inscrições [recebem US\\$ 400](#) para gastar no Confluent Cloud durante os primeiros 30 dias. Para evitar ter que inserir um cartão de crédito, navegue até [Faturamento e pagamento](#), role até o final e adicione o código promocional **CONFLUENTDEV1**. Com este código promocional, você não precisará inserir as informações do seu cartão de crédito por 30 dias ou até que seus créditos acabem.

Se você preferir configurar um cluster Kafka local, o tutorial também o guiará por essas etapas.

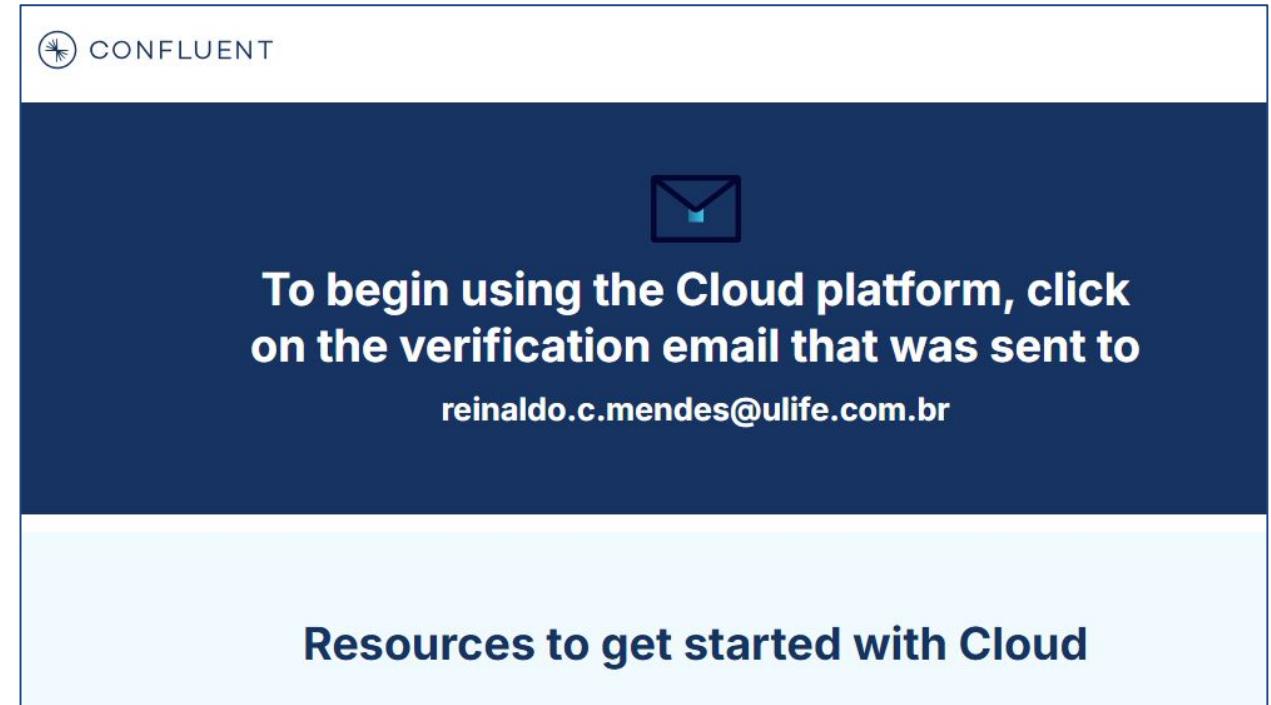
Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



The screenshot shows a verification email from Confluent. At the top left is the Confluent logo. In the center is a blue envelope icon with a small green checkmark. Below the icon, the text reads: "To begin using the Cloud platform, click on the verification email that was sent to". Underneath this, the recipient's email address is listed: "reinaldo.c.mendes@ulife.com.br". At the bottom of the email body, the text "Resources to get started with Cloud" is visible.

CONFLUENT

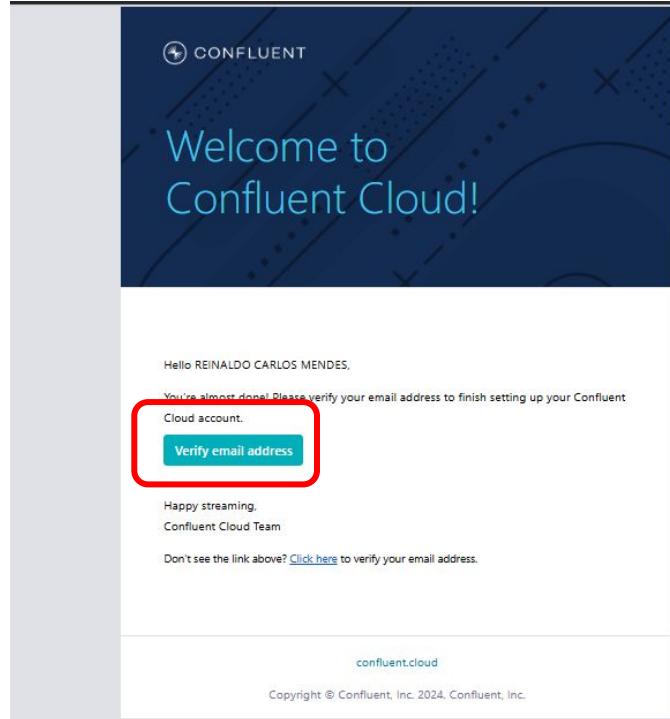
To begin using the Cloud platform, click on the verification email that was sent to

reinaldo.c.mendes@ulife.com.br

Resources to get started with Cloud

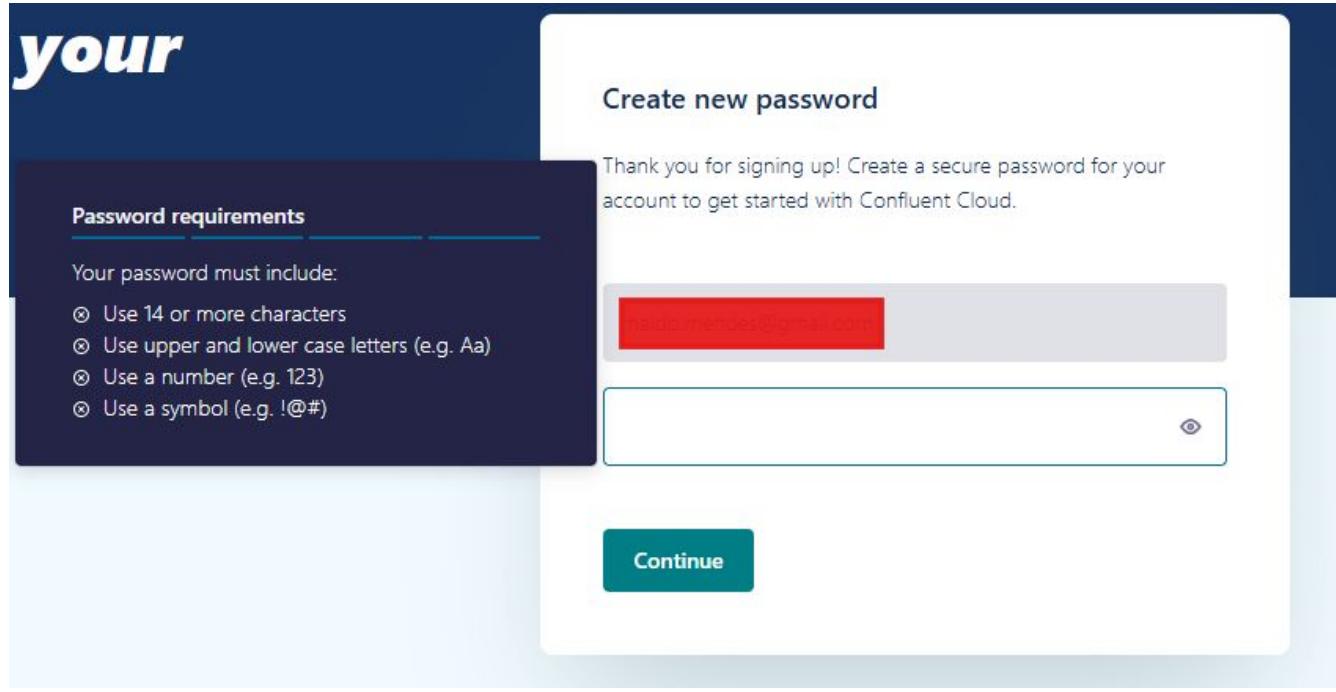
Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

Quão familiarizado você está com Kafka?*

Novo (menos de ~6 meses)

O que traz você ao Confluent Cloud?*

Explorando para aprendizagem pessoal

Você tem dados prontos para se conectar ao Confluent Cloud?*

Não, estou interessado em dados de amostra

Qual é o seu principal caso de uso com o Confluent Cloud?*

Microsserviços orientados a eventos

Você tem uma carga de trabalho Kafka existente para migrar para o Confluent Cloud?* ⓘ

Não

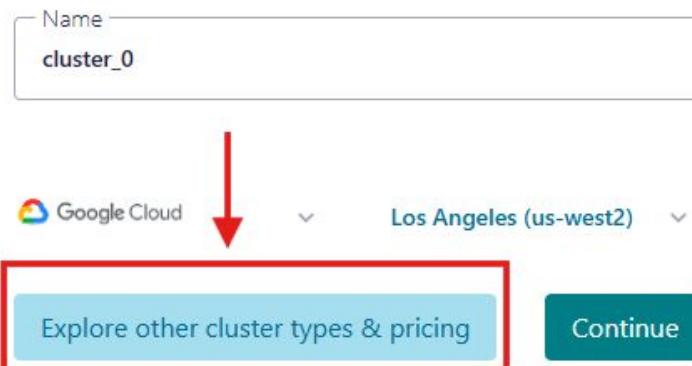
Próximo

Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

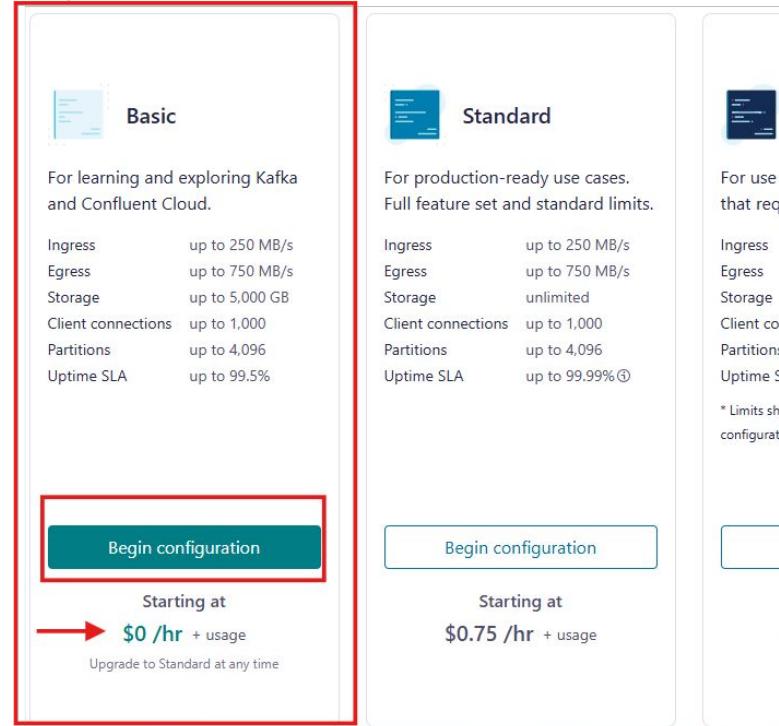
Create your first cluster!

Clusters store your data for real-time streaming and processing. Let's create your first Standard cluster, perfect for development and production - starting at \$0.75/hr + usage, with the Stream Governance Essentials Package included! ⓘ



Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



Plan	Ingress	Egress	Storage	Client connections	Partitions	Uptime SLA
Basic	up to 250 MB/s	up to 750 MB/s	up to 5,000 GB	up to 1,000	up to 4,096	up to 99.5%
Standard	up to 250 MB/s	up to 750 MB/s	unlimited	up to 1,000	up to 4,096	up to 99.99% ^④
Enterprise						

Basic
For learning and exploring Kafka and Confluent Cloud.

Ingress up to 250 MB/s
Egress up to 750 MB/s
Storage up to 5,000 GB
Client connections up to 1,000
Partitions up to 4,096
Uptime SLA up to 99.5%

Standard
For production-ready use cases. Full feature set and standard limits.

Ingress up to 250 MB/s
Egress up to 750 MB/s
Storage unlimited
Client connections up to 1,000
Partitions up to 4,096
Uptime SLA up to 99.99%^④

* Limits shown are for a single client connection. Configuration required.

Begin configuration

Starting at **\$0 /hr** + usage
Upgrade to Standard at any time

Start free

Standard
For use cases that require more resources.

Ingress up to 250 MB/s
Egress up to 750 MB/s
Storage unlimited
Client connections up to 1,000
Partitions up to 4,096
Uptime SLA up to 99.99%^④

* Limits shown are for a single client connection. Configuration required.

Begin configuration

Starting at **\$0.75 /hr** + usage

Enterprise
For use cases that require the highest level of performance and availability.

Ingress up to 250 MB/s
Egress up to 750 MB/s
Storage unlimited
Client connections up to 1,000
Partitions up to 4,096
Uptime SLA up to 99.99%^④

* Limits shown are for a single client connection. Configuration required.

Begin configuration

\$

Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

1. Cluster type —— 2. Region/zones 3. Payment 4. Review and launch



Region*

Los Angeles (us-west2)

Uptime SLA* ⓘ

99.5%

[Go back](#)

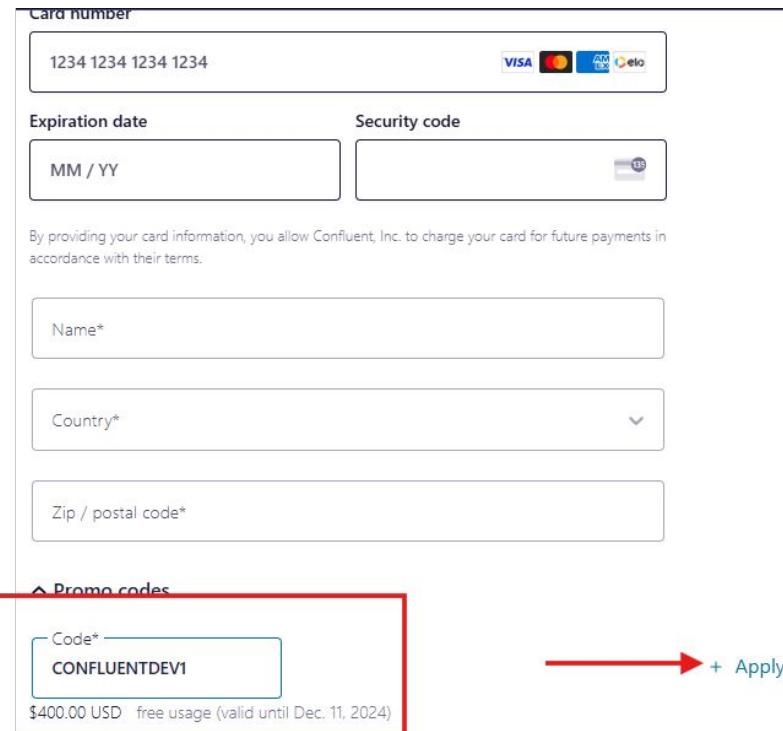
First eCKU is free, \$0.16 /hr after

Continue



Confluent Kafka Plataforma

- COLE O CÓDIGO PROMOCIONAL AQUI



The screenshot shows a payment form for Confluent Kafka Plataforma. At the bottom left, there's a section labeled "Promo codes" with a red rectangular box around it. Inside this box is a "Code*" input field containing the text "CONFLUENTDEV1". To the right of this input field is a red arrow pointing to a green "Apply" button. Below the input field, a note states "\$400.00 USD free usage (valid until Dec. 11, 2024)". The rest of the form includes fields for Card number, Expiration date, Security code, Name*, Country*, and Zip / postal code*.

Card number

1234 1234 1234 1234 VISA MASTERCARD AMEX ELO

Expiration date

MM / YY

Security code

Name*

Country*

Zip / postal code*

Promo codes

Code* CONFLUENTDEV1

\$400.00 USD free usage (valid until Dec. 11, 2024)

+ Apply

Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

Create cluster

1. Cluster type —— 2. Region/zones —— 3. Review and launch

Cluster name ⓘ

cluster_aula_sdm_2024

E-CKU cost First eCKU is free, \$0.15525 /hr after

Write \$0.0575 /GB

Read \$0.0575 /GB

Storage \$0.00012603 /GB-hour

Included: Stream Governance Essentials Package [Upgrade to Advanced](#)

Stream Governance is a fully managed data governance suite that will allow you to discover, understand, and trust your data streams. We have enabled the **Essentials Package** for you with GCP for free in the Los Angeles (us-west2) region.

[Go back](#)

[Launch cluster](#)

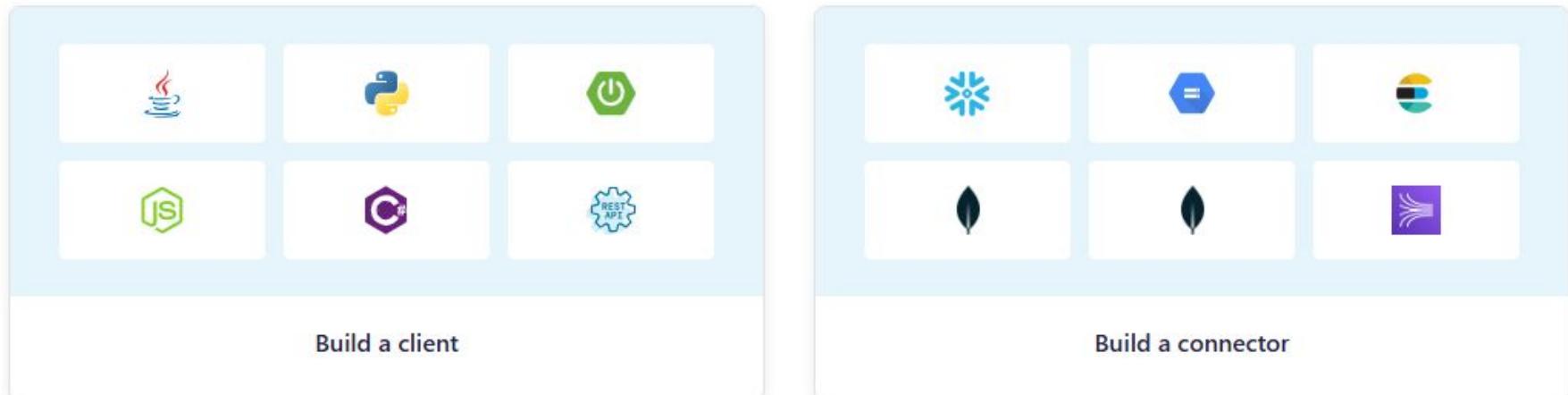
Confluent Kafka Plataforma

- Clique no seu cluster (**cluster_aula_sdm_2024**) e será direcionado para a cluster overview.

Welcome to Confluent Cloud, REINALDO CARLOS

🎉 Cluster **cluster_aula_sdm_2024** created!

What's next? Select one of the options to start generating data and develop your first pipeline.

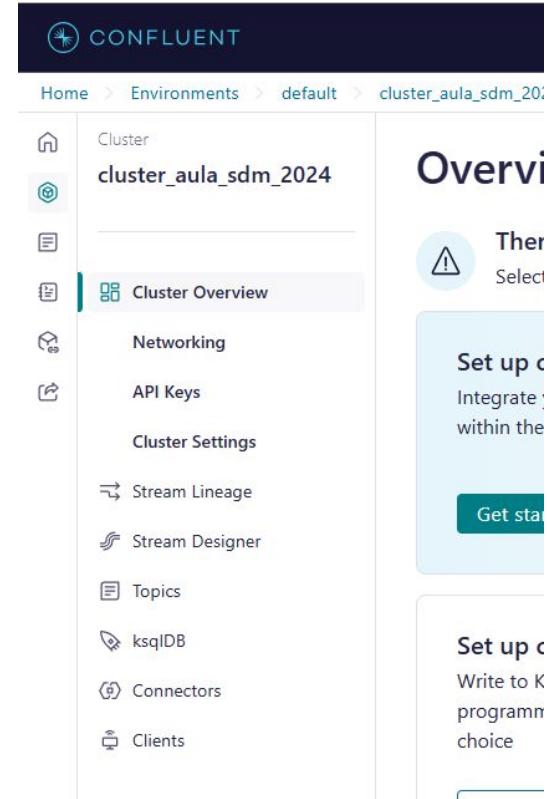


The screenshot shows the Confluent Cloud dashboard. At the top, a message indicates a new cluster ('cluster_aula_sdm_2024') has been created. Below this, there are two main sections:

- Build a client**: This section contains icons for various programming languages and tools: Java (coffee cup), Python (Python logo), Node.js (green power button), JavaScript (JS logo), C (C logo), and REST API (REST API logo). A large button at the bottom says "Build a client".
- Build a connector**: This section contains icons for data connectors: Flink (blue snowflake), Kafka Connect (blue hexagon with equals sign), Confluent Platform (yellow and green gear), and Apache Avro (purple leaf-like icon). A large button at the bottom says "Build a connector".

Confluent Kafka Plataforma

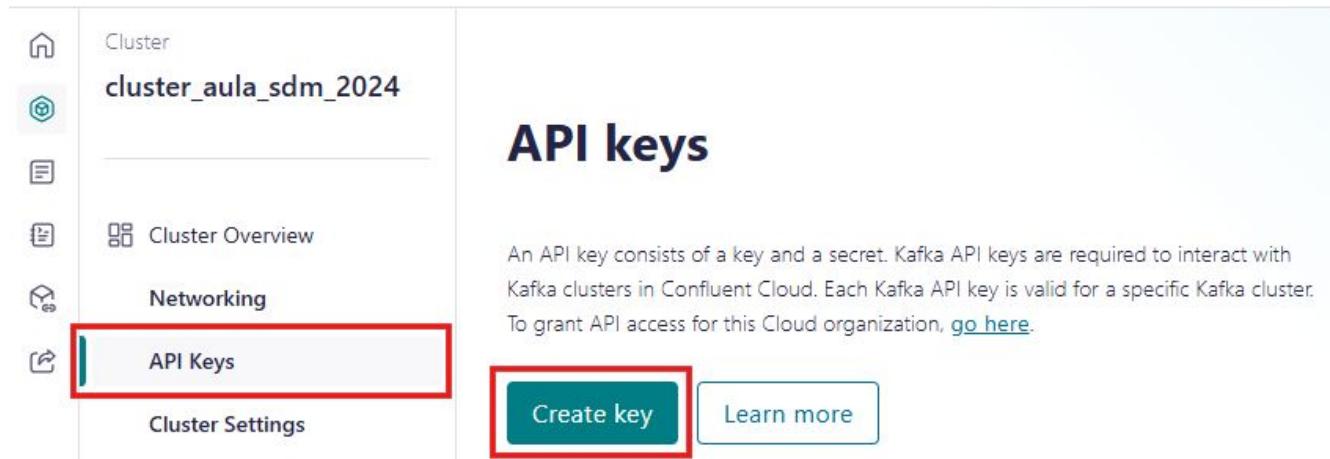
- Clique no seu Cluster Overview



The screenshot shows the Confluent Platform interface. At the top, there's a dark header with the CONFLUENT logo. Below it, a breadcrumb navigation shows: Home > Environments > default > cluster_aula_sdm_2024. The main area has a sidebar on the left with icons for Home, Environment, Cluster, Topic, API Keys, Cluster Settings, Stream Lineage, Stream Designer, Topics, ksqlDB, Connectors, and Clients. The 'Cluster' icon is selected, and its details are shown in the main content area: 'cluster_aula_sdm_2024'. To the right, there are two large callout boxes. The top one is titled 'Overview' and contains sections for 'Thermal Protection' (with an exclamation icon) and 'Select a cluster'. The bottom one is titled 'Set up cluster' and contains sections for 'Integrate with your data pipeline' (with a plus icon) and 'Get started' (with a green button). Both boxes have a light blue background.

Confluent Kafka Plataforma

- Clique em API Keys, para criar suas credenciais de acesso ao Cluster. Em seguida clique em Create Key



Cluster

cluster_aula_sdm_2024

Cluster Overview

Networking

API Keys

Cluster Settings

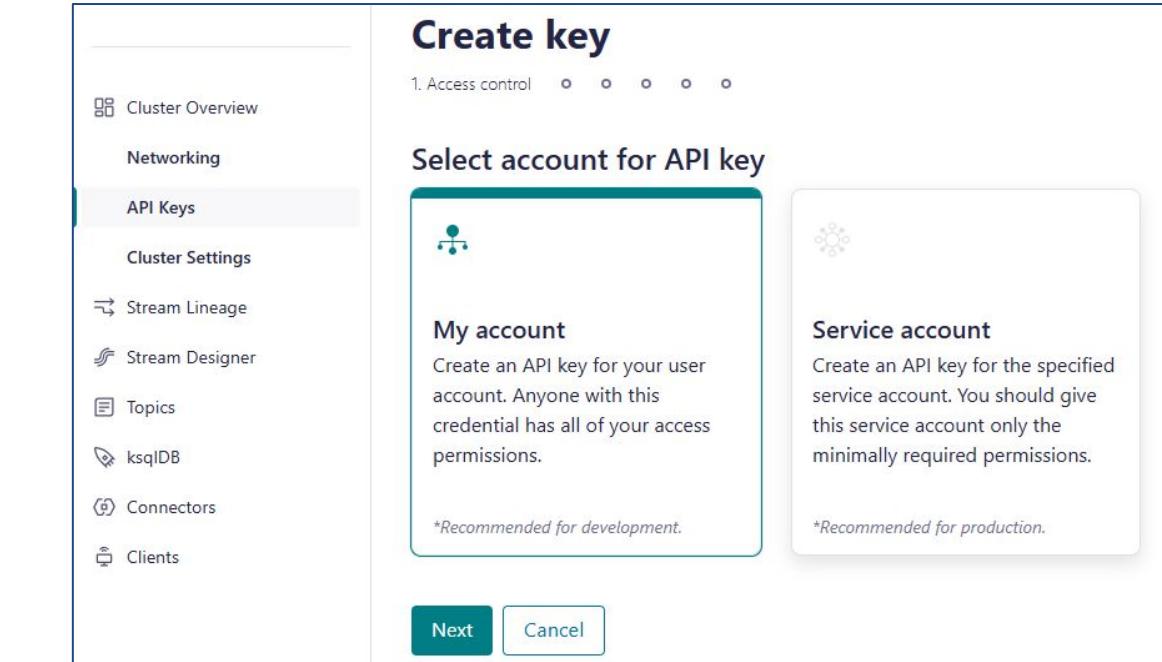
API keys

An API key consists of a key and a secret. Kafka API keys are required to interact with Kafka clusters in Confluent Cloud. Each Kafka API key is valid for a specific Kafka cluster. To grant API access for this Cloud organization, [go here](#).

Create key Learn more

Confluent Kafka Plataforma

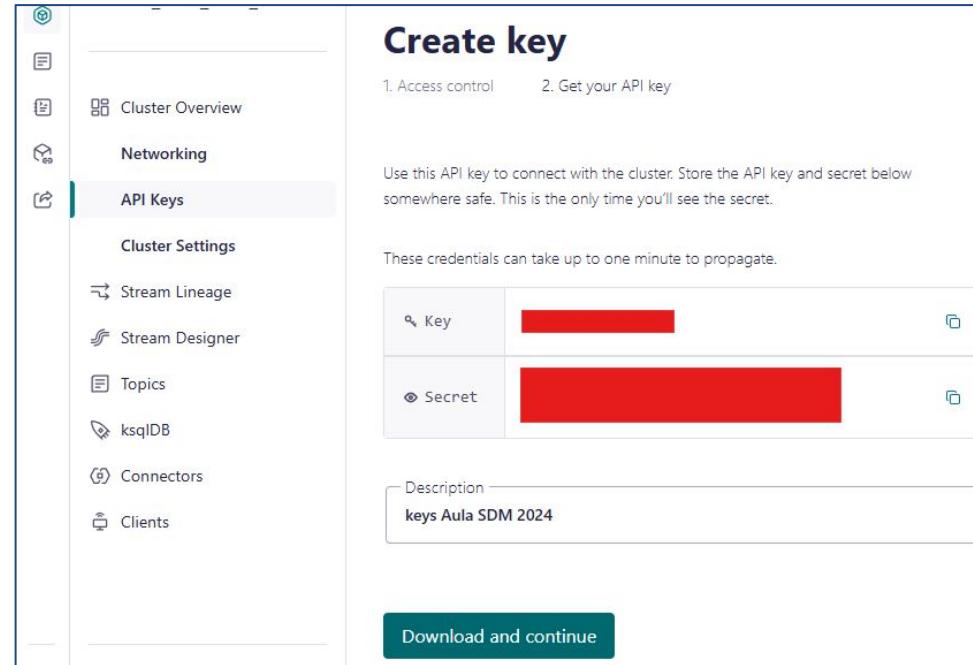
- Para experimentação, crie uma API Key para desenvolvimento. “My account”



The screenshot shows the 'Create key' process in the Confluent Kafka Platform. On the left, a sidebar menu includes 'Cluster Overview', 'Networking', 'API Keys' (which is selected), 'Cluster Settings', 'Stream Lineage', 'Stream Designer', 'Topics', 'ksqlDB', 'Connectors', and 'Clients'. The main area is titled 'Create key' and shows step 1: 'Access control' (with five dots indicating progress). Below this, a section titled 'Select account for API key' offers two options: 'My account' (selected) and 'Service account'. The 'My account' section contains the text: 'Create an API key for your user account. Anyone with this credential has all of your access permissions.' and a note: '*Recommended for development.' The 'Service account' section contains the text: 'Create an API key for the specified service account. You should give this service account only the minimally required permissions.' and a note: '*Recommended for production.' At the bottom are 'Next' and 'Cancel' buttons.

Confluent Kafka Plataforma

- Faça o download da chave criada e salve em um lugar seguro. precisaremos dela futuramente.



Confluent Kafka Plataforma

- Vamos criar um novo tópico no kafka.

The screenshot shows the Confluent Platform interface for a cluster named 'cluster_aula_sdm_2024'. The left sidebar lists various cluster management options: Cluster Overview, Networking, API Keys, Cluster Settings, Stream Lineage, Stream Designer, Topics (which is selected and highlighted with a red oval), ksqlDB, Connectors, and Clients. The main content area is titled 'Topics' and contains a descriptive paragraph about what a topic is. A prominent blue button labeled 'Create topic' is circled with a red oval. Below this, there's a 'Resources' section featuring a video thumbnail for 'Apache Kafka 101: Topics' with the caption 'Watch a video on how you can store and organize your events in a kafka topic.' and a 'Watch video →' link. To the right, there are partially visible sections for 'Documentation' and 'Read topics'.

Confluent Kafka Plataforma

- Neste momento não utilizaremos opções avançadas.

New topic

Topic name* ⓘ — vendas_online

Partitions* ⓘ — 3

Enable infinite retention for this topic

Retain your data in Kafka infinitely to access historical and current data all in one place for data analysis, regulatory purposes, or other use cases.
[Learn more ↗](#)

Show advanced settings

Confluent Kafka Plataforma

- Neste momento NÃO adicionaremos nenhum contrato. clique em **pular** ou **skip**

Seu tópico foi criado com sucesso!

Contrato de dados

Estabelecer contratos de dados é essencial para garantir estruturas de dados uniformes, validações e políticas em sua plataforma de streaming de dados. Essa prática permite governança eficaz para o crescimento da plataforma. Opte por esquemas Avro, Protobuf ou JSON para estabelecer uma estrutura coesa que promova comunicação contínua entre seus vários aplicativos e serviços. [Saiba mais ↗](#)

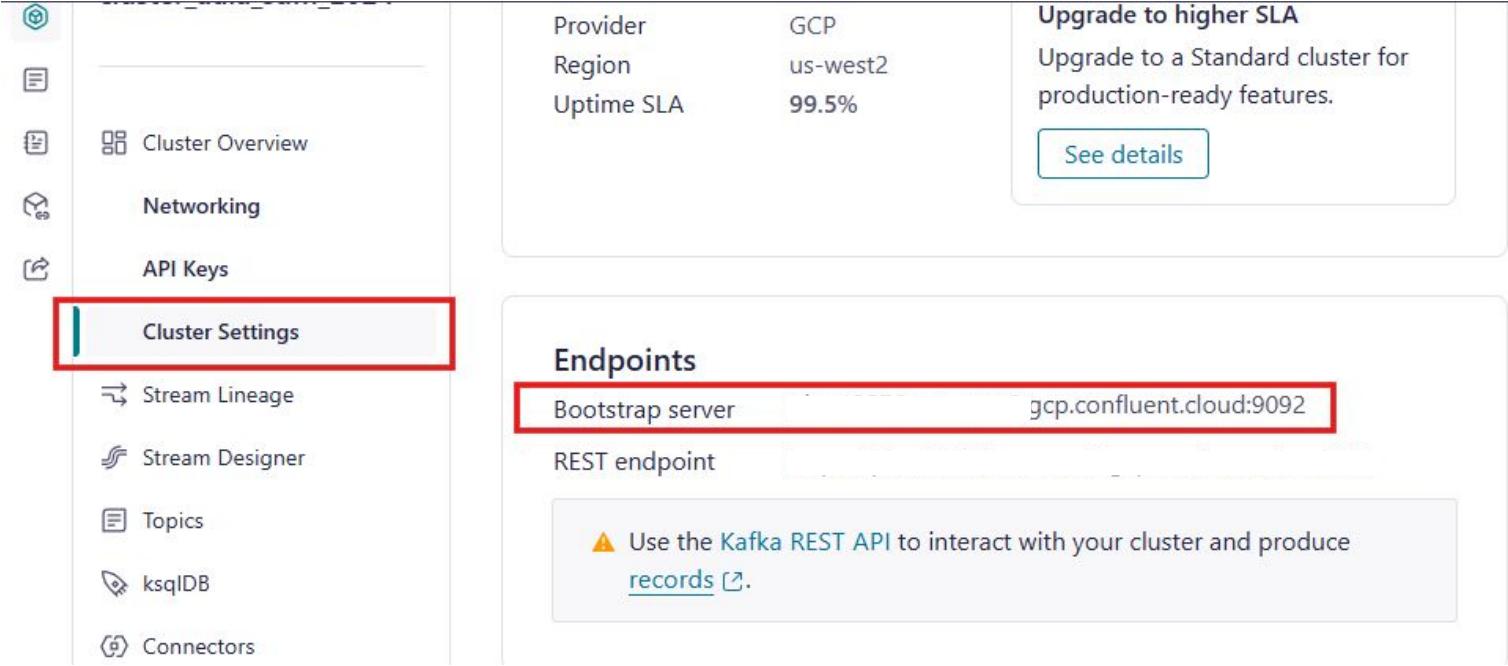
- Adicionar contrato de dados para os valores da mensagem do tópico ① Mais comum
- Adicionar contrato de dados para as chaves de mensagem do tópico ①

Pular

Adicionar contrato de dados

Confluent Kafka Plataforma

- Em cluster configuração, temos o link bootstrap para o servidor Kafka



Provider: GCP
Region: us-west2
Uptime SLA: 99.5%

Upgrade to higher SLA
Upgrade to a Standard cluster for production-ready features.
[See details](#)

Endpoints

Bootstrap server	gcp.confluent.cloud:9092
REST endpoint	

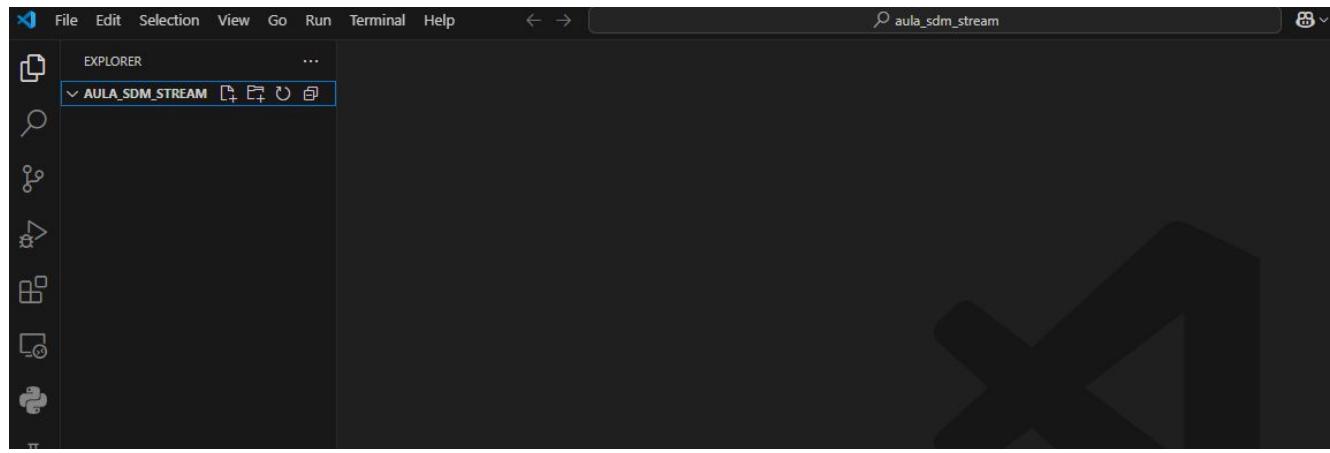
⚠ Use the [Kafka REST API](#) to interact with your cluster and produce [records](#).

- Cluster Overview
- Networking
- API Keys
- Cluster Settings**
- Stream Lineage
- Stream Designer
- Topics
- ksqlDB
- Connectors

Confluent Kafka Plataforma

Agora vamos criar um novo projeto python.

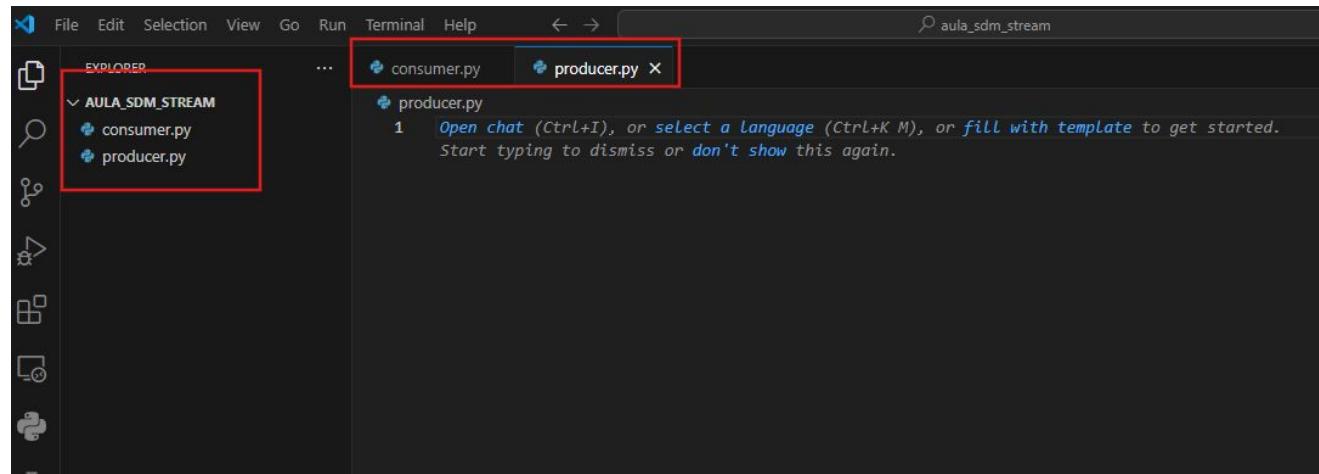
1. Crie um pasta chamada **aula_sdm_stream** em um lugar de sua escolha no computador.
2. Dentro do VS Code abra a pasta que você criou.



Confluent Kafka Plataforma

Crie os arquivos para produzir e consumir itens do Kafka.

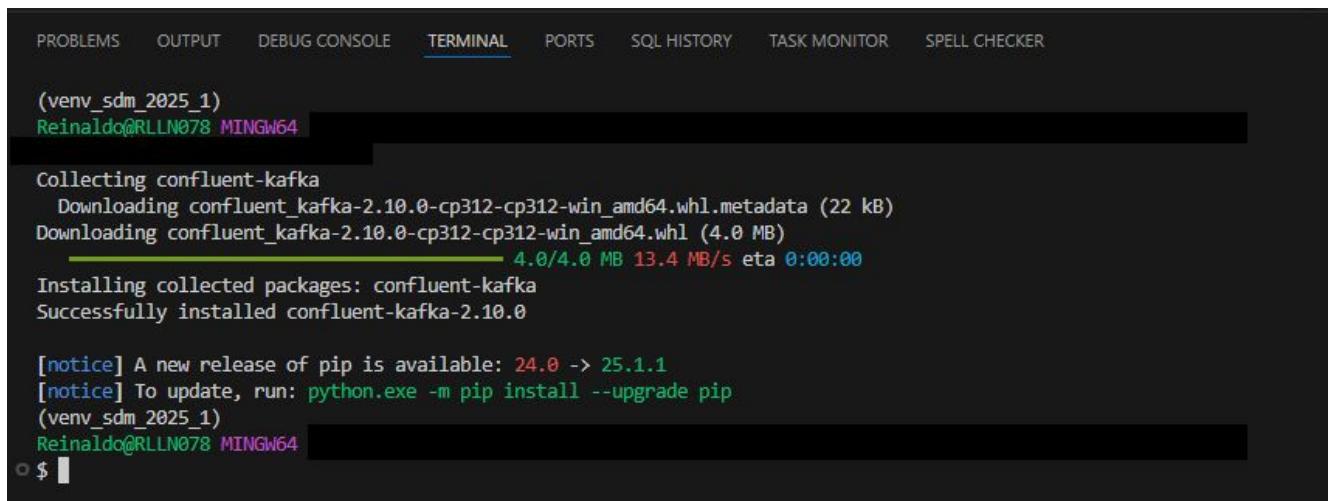
3. Crie os arquivos consumer.py e producer.py dentro do seu projeto.



Confluent Kafka Plataforma

Instale a biblioteca do confluent

4. Instale o biblioteca do confluent-kafka no seu projeto.
 - a. pip install confluent-kafka



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR SPELL CHECKER

(venv_sdm_2025_1)
Reinaldo@RLLN078 MINGW64

Collecting confluent-kafka
  Downloading confluent_kafka-2.10.0-cp312-cp312-win_amd64.whl.metadata (22 kB)
  Downloading confluent_kafka-2.10.0-cp312-cp312-win_amd64.whl (4.0 MB)
    4.0/4.0 MB 13.4 MB/s eta 0:00:00
Installing collected packages: confluent-kafka
Successfully installed confluent-kafka-2.10.0

[notice] A new release of pip is available: 24.0 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv_sdm_2025_1)
Reinaldo@RLLN078 MINGW64
$
```

Confluent Kafka Plataforma

**Edite o arquivo
PRODUCER.PY**

producer.py ●

producer.py > ...

```
1  from random import choice
2  from confluent_kafka import Producer
3
4  if __name__ == '__main__':
5
6      config = {
7          # Servidor BootStrap, Usuário e Senha
8          'bootstrap.servers': '',
9          'sasl.username': '',
10         'sasl.password': '',
11
12         # Propriedades Padrão do Confluent
13         'security.protocol': 'SASL_SSL',
14         'sasl.mechanisms': 'PLAIN',
15
16         # Tempo de espera de resposta antes de cancelar a conexão
17         'session.timeout.ms': 45000
18     }
19
20     # Cria uma instância de produtor
21     producer = Producer(config)
22
23     # Nome do Tópico
24     topic = "NOME_TOPICO"
25
26     # Função opcional de devolução.
27     # É chamada por poll() e flush()
28     def delivery_callback(err, msg):
29         if err:
30             print('ERROR: Message failed delivery: {}'.format(err))
```

producer.py

producer.py > ...

```
25
26     # Função opcional de devolução.
27     # É chamada por poll() e flush()
28     def delivery_callback(err, msg):
29         if err:
30             print('ERROR: Message failed delivery: {}'.format(err))
31         else:
32             print("Produced event to topic {topic}: key = {key:12} value = {value:12}".format(
33                 topic=msg.topic(), key=msg.key().decode('utf-8'), value=msg.value().decode('utf-8')))
34
35     # Produz dados combinando valores das listas aleatoriamente.
36     user_ids = ['psilva', 'smanoel', 'jpires', 'jbernardo', 'hmoraes', 'fmilagres']
37     products = ['Camisa do Galo', 'Rádio Relogio', 'Bermuda Praia', 'Mouse', 'Teclado']
38
39     count = 0
40     for _ in range(10):
41         user_id = choice(user_ids)
42         product = choice(products)
43         producer.produce(topic, product, user_id, callback=delivery_callback)
44         count += 1
45
46     # Block until the messages are sent.
47     producer.poll(10000)
48     producer.flush()
```

Confluent Kafka Plataforma

Definindo servidor, usuário e senha

5. Modifique as informações de servidor usuário e senha no seu código para acessar o kafka. Esses dados estão no arquivo salvo por você e na plataforma do confluent.
 - 5.1. *API key (usuário)*
 - 5.2. *API secret (senha)*
 - 5.3. *Bootstrap server (host Servidor Kafka)*
6. Altere o nome do Tópico para o nome que você criou no Kafka (vendas_online)

producer.py X

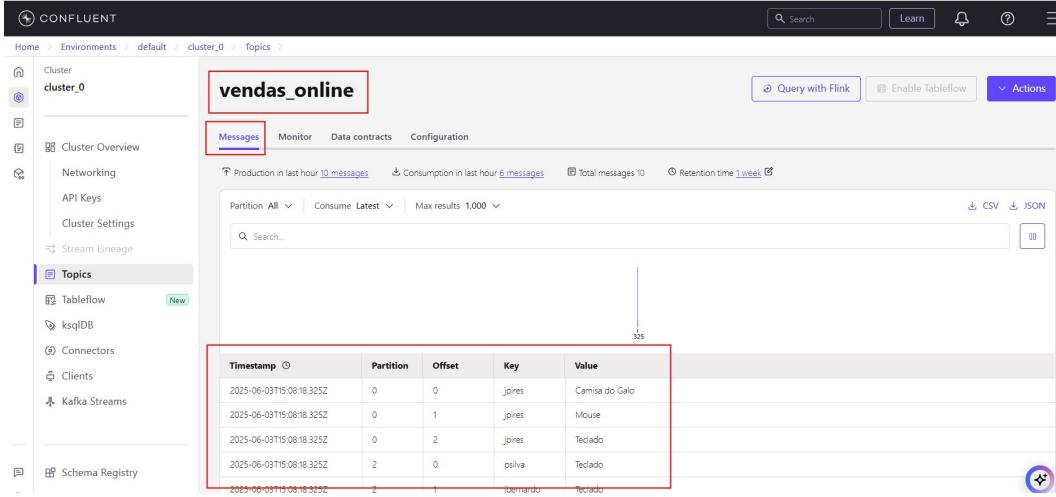
producer.py > ...

```
3
4  if __name__ == '__main__':
5
6      config = [
7          # Servidor BootStrap, Usuário e Senha
8          'bootstrap.servers': 'PLAINTEXT://ec2-18-215-111-111.confluent.cloud:9092',
9          'sasl.username': 'GSX7E6EXJ5N',
10         'sasl.password': 'V77d22jYxBhCEIqog9U073',
11
12         # Propriedades Padrão do Confluent
13         'security.protocol': 'SASL_SSL',
14         'sasl.mechanisms': 'PLAIN',
15
16         # Tempo de espera de resposta antes de cancelar a conexão
17         'session.timeout.ms': 45000
18     ]
19
20     # Cria uma instância de produtor
21     producer = Producer(config)
22
23     # Nome do Tópico
24     topic = "vendas_online"
```

Confluent Kafka Plataforma

Executando Producer.py

7. Execute seu código e observe se os eventos foram submetidos no tópico do Kafka.
 - 7.1. Entre no Tópico na plataforma do Confluent e verifique se os eventos estão disponíveis no Tópico.



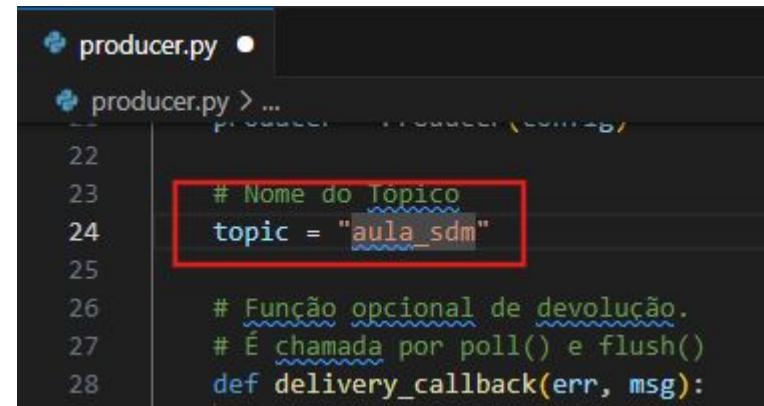
The screenshot shows the Confluent Platform interface. On the left, there's a sidebar with options like Cluster Overview, API Keys, Cluster Settings, Stream Lineage, Topics (which is selected), Tableflow, ksqlDB, Connectors, Clients, Kafka Streams, and Schema Registry. The main area is titled 'vendas_online'. It has tabs for Messages (selected), Monitor, Data contracts, and Configuration. Below the tabs, it shows production and consumption metrics: Production in last hour (10 messages), Consumption in last hour (6 messages), Total messages (10), and Retention time (1 week). A search bar and CSV/JSON download buttons are also present. The main content area displays a table of messages with columns: Timestamp, Partition, Offset, Key, and Value. The table shows four rows of data:

Timestamp	Partition	Offset	Key	Value
2025-06-03T15:08:18.325Z	0	0	jpires	Camisa do Galo
2025-06-03T15:08:18.325Z	0	1	jpires	Mouse
2025-06-03T15:08:18.325Z	0	2	jpires	Tecelado
2025-06-03T15:08:18.325Z	2	0	psilva	Tecelado
2025-06-03T15:08:18.325Z	2	1	lfernando	Tecelado

Confluent Kafka Plataforma

Mensagem de retorno para erro (Callback message)

8. A função `delivery_callback` tem o objetivo de enviar uma mensagem para o sistema chamador, quando ocorre um erro ao tentar postar um evento em um tópico do kafka.
 - 8.1. Gere um distúrbio no código para visualizamos a função de callback em operação.
 - 8.1.1. altere o nome do tópico no código e execute `produtor.py` novamente.



```
producer.py •
producer.py > ...
22
23
24 # Nome do Tópico
topic = "aula_sdm"
25
26 # Função opcional de devolução.
27 # É chamada por poll() e flush()
28 def delivery_callback(err, msg):
```

Confluent Kafka Plataforma

**Edite o arquivo
CONSUMER.PY**

Confluent Kafka Plataforma

```
❸ consumer.py •
❹ consumer.py > ...
1   from confluent_kafka import Consumer
2
3   if __name__ == '__main__':
4
5       config = {
6           # Servidor Bootstrap, Usuário e Senha
7           'bootstrap.servers': '',
8           'sasl.username': '',
9           'sasl.password': '',
10
11           # Propriedades Padrão do Confluent
12           'security.protocol': 'SASL_SSL',
13           'sasl.mechanisms': 'PLAIN',
14
15           # Tempo de espera de entre cada tentativa de consumir um mensagem
16           'session.timeout.ms': 45000,
17
18           # Identificador do ID Grupo
19           'group.id': 'kafka-python-aprendendo',
20
21           # Configuração padrão para recuperacão
22           'auto.offset.reset': 'earliest'
23
24 }
```

Confluent Kafka Plataforma

```
25      # Cria uma instância de Consumidor
26      consumer = Consumer(config)
27
28      # Inscrição em um Tópico
29      topic = "NOME_TOPICO"
30      consumer.subscribe([topic])
31
32      # Poll para novas mensagens do Kafka e Imprimindo Elas.
33      try:
34          while True:
35              msg = consumer.poll(1.0)
36              if msg is None:
37                  print("Aguardando...")
38              elif msg.error():
39                  print("ERROR: {}".format(msg.error()))
40              else:
41                  # Extrai Key e Value (Chave e Valor) e Imprimi. Faz a decodificação da Mensagem.
42                  print("Consumindo Evento do Tópico {}: chave = {} Valor = {}".format(
43                      topic=msg.topic(), key=msg.key().decode('utf-8'), value=msg.value().decode('utf-8')))
44      except KeyboardInterrupt:
45          pass
46      finally:
47          # Deixa o Grupo de commit offsets finais
48          consumer.close()
```

Confluent Kafka Plataforma

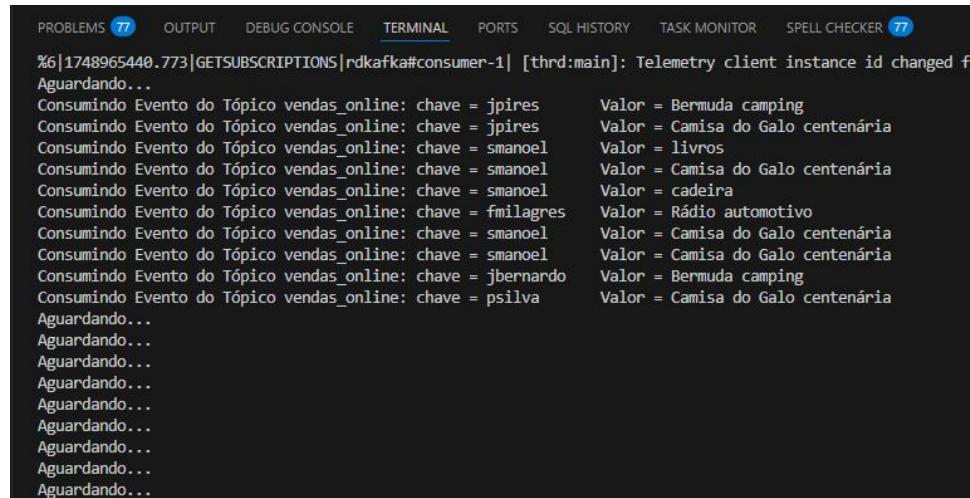
Definindo servidor, usuário e senha

9. Modifique as informações de servidor usuário e senha no seu código para acessar o kafka. Esses dados estão no arquivo salvo por você e na plataforma do confluent.
 - 9.1. Realize o mesmo procedimento executado para o producer.py
10. Altere o nome do Tópico para o nome que você criou no Kafka (vendas_online)

Confluent Kafka Plataforma

Executando Producer.py

11. Execute seu código e observe se os eventos disponíveis no tópico do Kafka, são consumidos pelo código do consumir.py
 - 11.1. Esse código continua rodando aguardando que o produtor.py gere novos eventos a serem consumidos.



The screenshot shows a terminal window with the following text output:

```
%6|1748965440.773|GETSUBSCRIPTIONS|rdkafka#consumer-1| [thrd:main]: Telemetry client instance id changed from 0 to 1
Aguardando...
Consumindo Evento do Tópico vendas_online: chave = jpires          Valor = Bermuda camping
Consumindo Evento do Tópico vendas_online: chave = jpires          Valor = Camisa do Galo centenária
Consumindo Evento do Tópico vendas_online: chave = smanoel           Valor = livros
Consumindo Evento do Tópico vendas_online: chave = smanoel           Valor = Camisa do Galo centenária
Consumindo Evento do Tópico vendas_online: chave = smanoel           Valor = cadeira
Consumindo Evento do Tópico vendas_online: chave = fmilagres          Valor = Rádio automotivo
Consumindo Evento do Tópico vendas_online: chave = smanoel           Valor = Camisa do Galo centenária
Consumindo Evento do Tópico vendas_online: chave = smanoel           Valor = Camisa do Galo centenária
Consumindo Evento do Tópico vendas_online: chave = jbernardo          Valor = Bermuda camping
Consumindo Evento do Tópico vendas_online: chave = psilva            Valor = Camisa do Galo centenária
Aguardando...
Aguardando...
Aguardando...
Aguardando...
Aguardando...
Aguardando...
Aguardando...
Aguardando...
Aguardando...
```

Confluent Kafka Plataforma

Em conjunto com um colega vocês vão realizar os passos de produtor e consumidor.

1. Pegue com um colega as credenciais necessárias para conectar ao servidor com o Kafka que ele criou.
2. Configure seu código de consumidor para consumir os eventos gerados pelo seu colega.
3. Execute seu código e deixe ele em execução e observe o terminal.
4. Peça ao seu colega para gerar novos usuários e produtos no código do producer.py dele.
5. Veja no seu terminal o que ocorre
6. Troque os papéis com seu colega, agora você é o produtor e ele o consumidor.

Referências

- OLIVEIRA NETO, Jesus José de. Aula 11-12. PUC Goiás, 2021. Disponível em: <https://professor.pucgoias.edu.br/sitedocente/admin/arquivosUpload/17510/material/Aula11-12.pdf>. Acesso em: 2 jun. 2024.
- ESTRELLA, Julio Cesar. Aula 10. ICMC, 2021. Disponível em: https://edisciplinas.usp.br/pluginfile.php/7749153/mod_page/content/17/Aula-10.zip. Acesso em: 1 jun. 2024.
- CARMO, Leo. Explorando os conceitos e diferenças entre mensageria e event streaming. 2021. Disponível em: <https://leocarmo.dev/explorando-os-conceitos-e-diferencias-entre-mensageria-e-event-streaming>. Acesso em: 1 ago. 2024.
- FIDELISSAURO. Mensageria, eventos e streaming. 2021. Disponível em: <https://fidelissauro.dev/mensageria-eventos-streaming/>. Acesso em: 1 ago. 2024.