

---

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 o 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\\_br/streams/latest/dev/using-other-services-read-kafka.html](https://docs.aws.amazon.com/pt_br/streams/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](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**

**Confluent, for free**

**New signups receive \$400 to spend during their first 30 days.**

You're minutes away from connecting, processing, and governing your streaming data on the only fully managed Apache Kafka® service reinvented for the hybrid cloud enterprise. Start your 30-day trial of Confluent Cloud today.

- ✓ **Cloud Native:** Stream data at massive scale without the operational burden or expense of open source software.
- ✓ **Complete:** Go way beyond Kafka to build real-time apps quickly, reliably, and securely
- ✓ **Everywhere:** Maintain flexibility with deployments across clouds and on-premises that sync in real time

**Use your cloud marketplace account to get started with Confluent Cloud.**

Microsoft Azure | aws marketplace | Google Cloud

**Sign In With Google**

**Sign In With GitHub**

OR

Reinaldo Carlos Mendes

Anima

reinaldo.c.mendes@ulife.com.br

Brazil

☐ Email me about products, services & events from Confluent

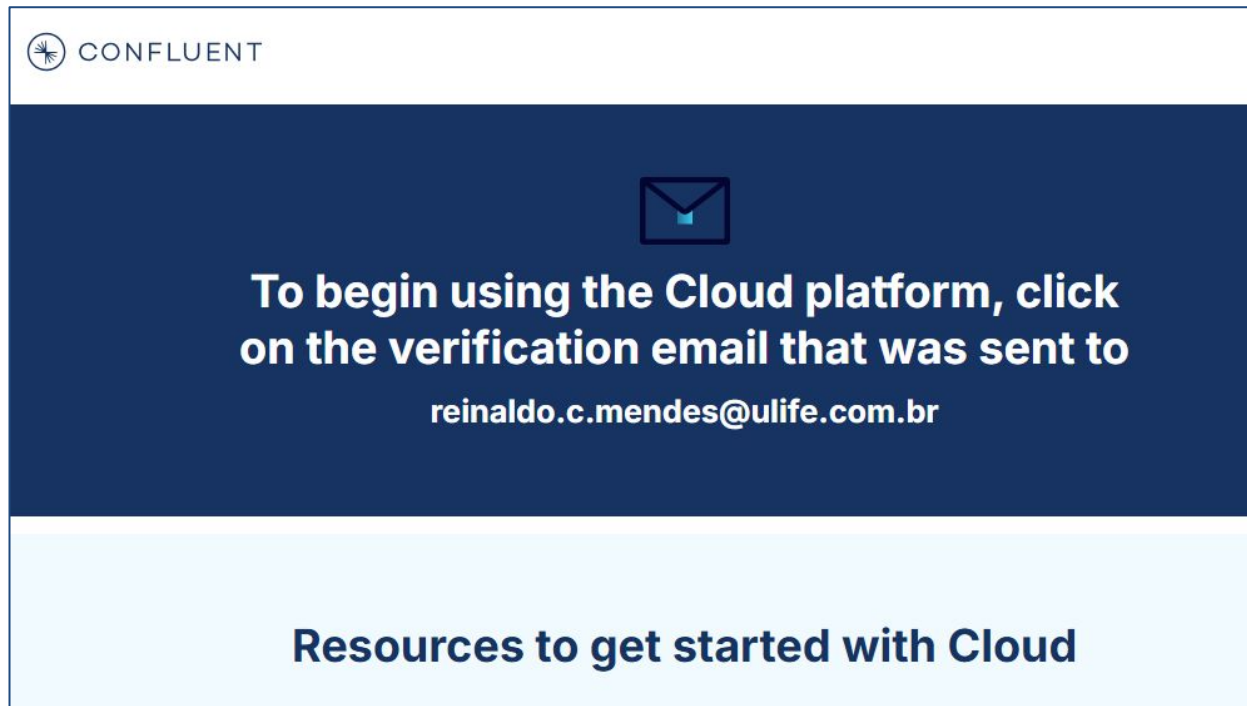
**Start Free**

I have an account. [Log-In](#)

By clicking "START FREE" above, you agree to the [Terms of Service](#) and [Privacy Policy](#).

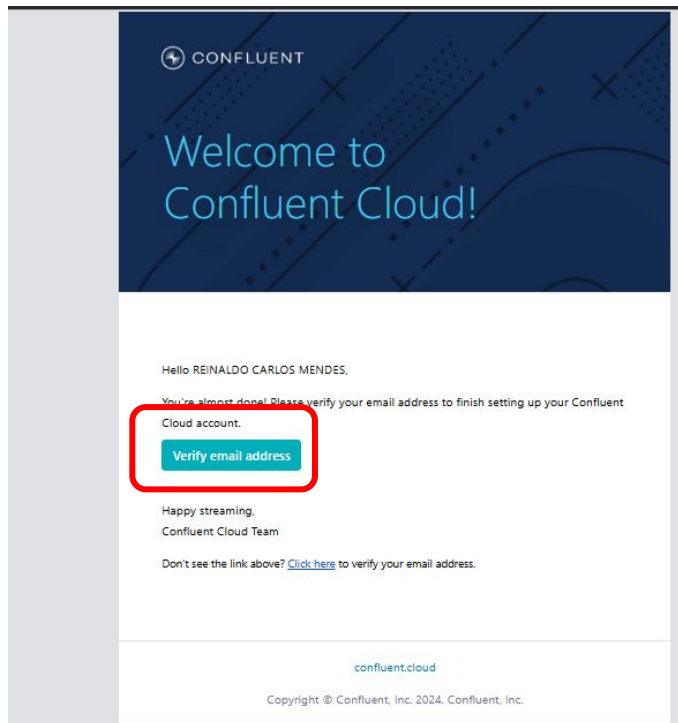
# Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



# Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma



# Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

**your**

**Password requirements**

Your password must include:

- ⊗ Use 14 or more characters
- ⊗ Use upper and lower case letters (e.g. Aa)
- ⊗ Use a number (e.g. 123)
- ⊗ Use a symbol (e.g. !@#)

**Create new password**

Thank you for signing up! Create a secure password for your account to get started with Confluent Cloud.

maria.mendes@confluent.com

Continue

# 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?\*

Microserviços 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! ⓘ

Name  
cluster\_0

Google Cloud ▼ Los Angeles (us-west2) ▼

Explore other cluster types & pricing Continue

# Confluent Kafka Plataforma

- Cadastrando gratuitamente na plataforma

The screenshot displays the Confluent Kafka pricing page with three plans: Basic, Standard, and Enterprise. The Basic plan is highlighted with a red box. Below the Basic plan, a red arrow points to the price '\$0 /hr + usage'.

| Plan       | Description   | Ingress        | Egress         | Storage        | Client connections | Partitions  | Uptime SLA     | Starting at        |
|------------|---|----------------|----------------|----------------|--------------------|-------------|----------------|--------------------|
| Basic      | For learning and exploring Kafka and Confluent Cloud.                 | 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%    | \$0 /hr + usage    |
| Standard   | For production-ready use cases. Full feature set and standard limits. | up to 250 MB/s | up to 750 MB/s | unlimited      | up to 1,000        | up to 4,096 | up to 99.99% ③ | \$0.75 /hr + usage |
| Enterprise | For use cases that require advanced features and SLAs.                | up to 250 MB/s | up to 750 MB/s | unlimited      | up to 1,000        | up to 4,096 | up to 99.99% ③ | \$1.50 /hr + usage |

\* Limits shown are for a single 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**

Card number

1234 1234 1234 1234

VISA Mastercard American Express Elo

Expiration date

MM / YY

Security code

123

By providing your card information, you allow Confluent, Inc. to charge your card for future payments in accordance with their terms.

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.





Build a client

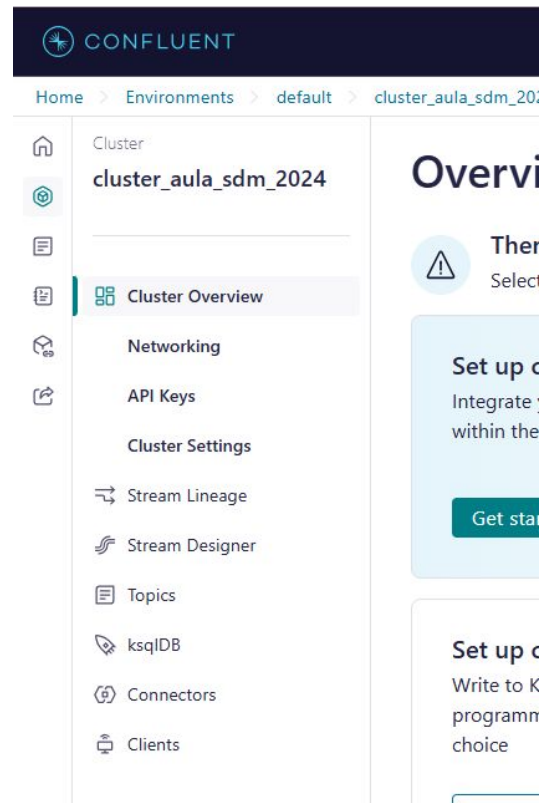




Build a connector

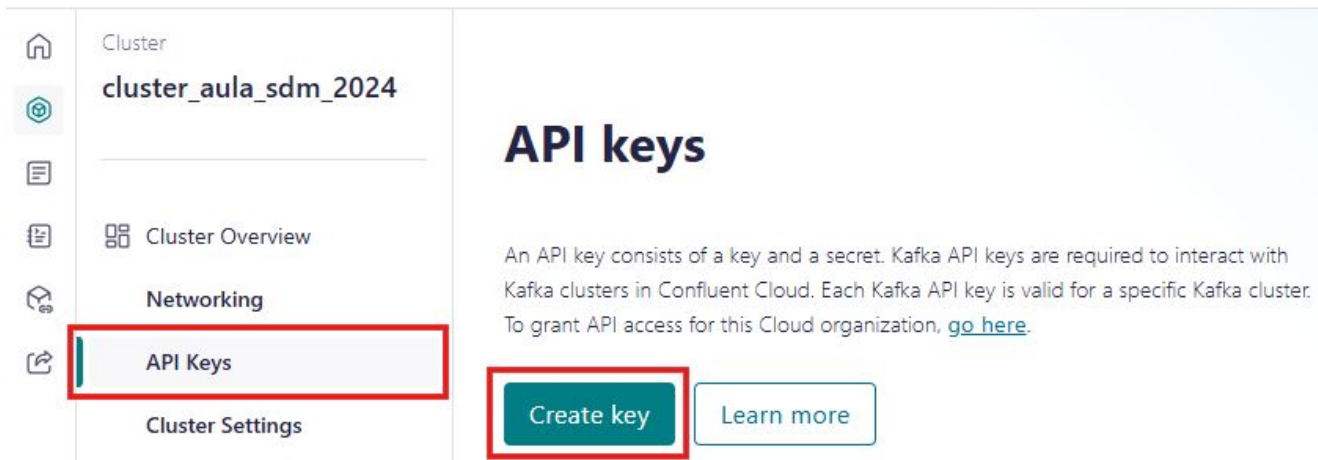
# Confluent Kafka Plataforma

- Clique no seu Cluster Overview



## Confluent Kafka Plataforma

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



# Confluent Kafka Plataforma

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

**Create key**

1. Access control ○ ○ ○ ○ ○

**Select account for API key**

**My account**  
Create an API key for your user account. Anyone with this credential has all of your access permissions.  
*\*Recommended for development.*

**Service account**  
Create an API key for the specified service account. You should give this service account only the minimally required permissions.  
*\*Recommended for production.*

**Next** **Cancel**

# Confluent Kafka Plataforma

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

**Create key**

1. Access control 2. Get your API key

Use this API key to connect with the cluster. Store the API key and secret below somewhere safe. This is the only time you'll see the secret.

These credentials can take up to one minute to propagate.

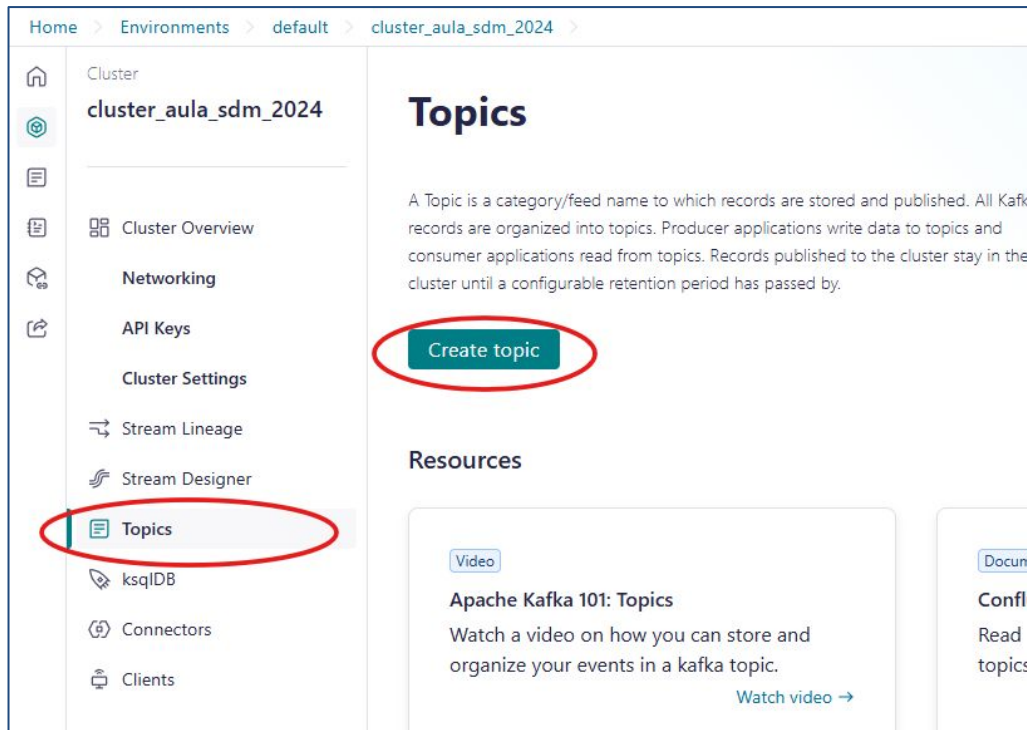
|        |            |      |
|--------|------------|------|
| Key    | [Redacted] | Copy |
| Secret | [Redacted] | Copy |

Description  
keys Aula SDM 2024

Download and continue

# Confluent Kafka Plataforma

- Vamos criar um novo tópico no kafka.



---

# 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](#)

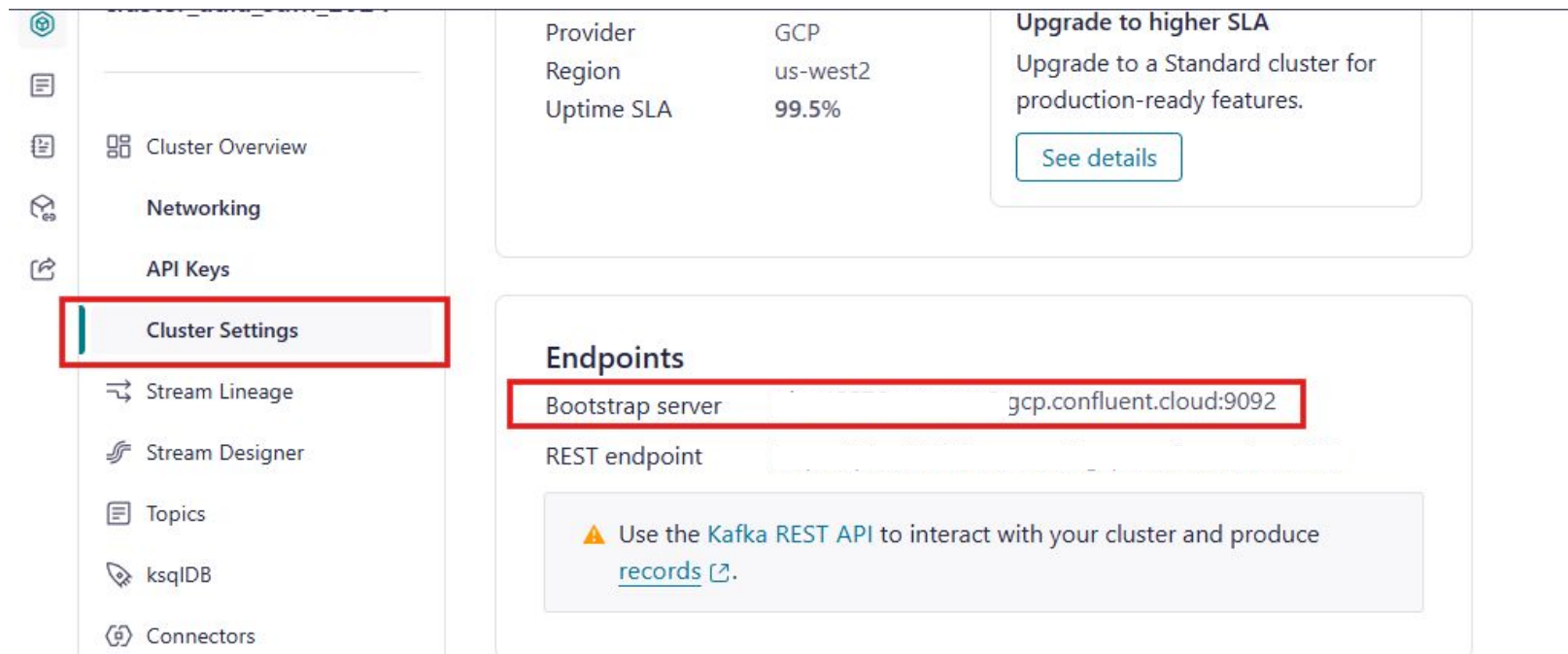
- ☒ Adicionar contrato de dados para os valores da mensagem do tópico ⓘ Mais comum
- ☐ Adicionar contrato de dados para as chaves de mensagem do tópico ⓘ

Pular

Adicionar contrato de dados

# Confluent Kafka Plataforma

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



The screenshot displays the Confluent Kafka Platform interface. On the left sidebar, the 'Cluster Settings' option is highlighted with a red box. The main content area shows the 'Cluster Overview' section with a table of cluster details:

| Provider   | GCP      | Upgrade to higher SLA  |
|------------|----------|--|
| Region     | us-west2 | Upgrade to a Standard cluster for production-ready features. |
| Uptime SLA | 99.5%    | <a href="#">See details</a>                                  |

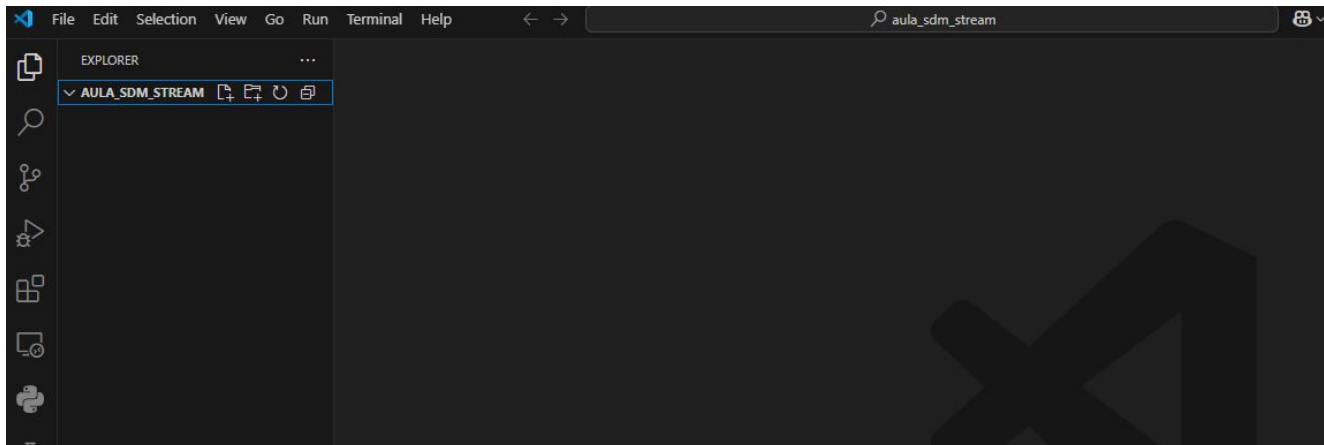
Below the table, the 'Endpoints' section is visible. The 'Bootstrap server' field is highlighted with a red box and contains the value 'gcp.confluent.cloud:9092'. The 'REST endpoint' field is also visible but empty. A note at the bottom states: 'Use the [Kafka REST API](#) to interact with your cluster and produce [records](#).'.

---

# Confluent Kafka Plataforma

Agora vamos criar um novo projeto python.

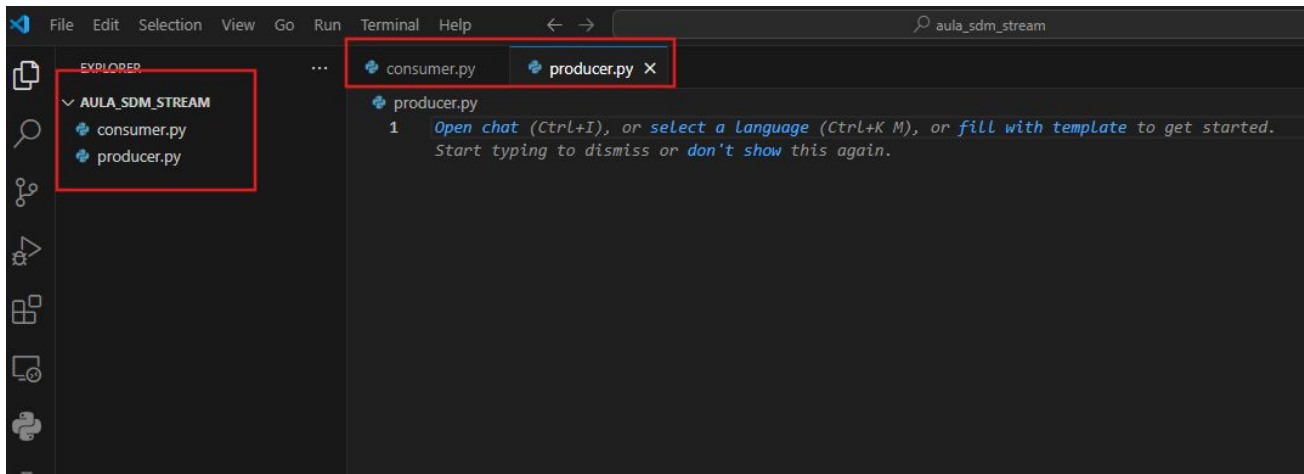
1. Crie um pasta chamada **aula\_sdm\_stream** em um lugar de sua escolha no computador.
2. Dentro do VS Code abra a pasta que você criou.



# 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@RLN078 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@RLN078 MINGW64
o $
```

# Edite o arquivo **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))

```

```
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 Relógio', '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)

```
3
4  if __name__ == '__main__':
5
6      config = {
7          # Servidor BootStrap, Usuário e Senha
8          'bootstrap.servers': '██████████.confluent.cloud:9092',
9          'sasl.username': 'GSX7E6EXJ!██████████N',
10         'sasl.password': 'V77d22jYxBhCE██████████Iqog9U073',
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"
25
```

# 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.

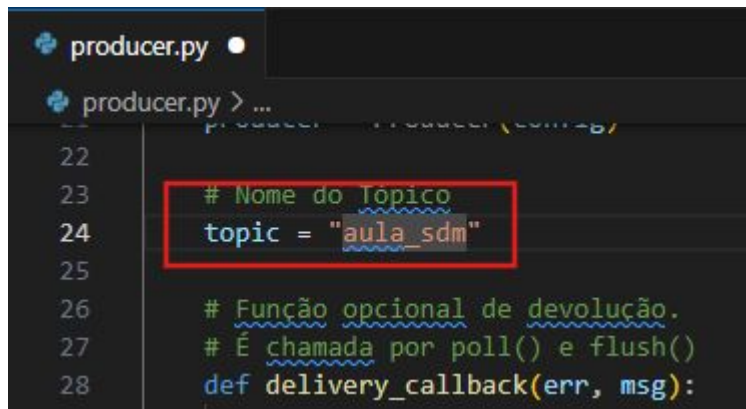
The screenshot shows the Confluent Kafka platform interface. The left sidebar contains navigation links for Cluster Overview, Networking, API Keys, Cluster Settings, Stream Lineage, Topics, Tableflow, ksqldb, Connectors, Clients, Kafka Streams, and Schema Registry. The main panel displays the 'vendas\_online' topic under 'cluster\_0'. The 'Messages' tab is selected, showing a table of messages. The table has columns: Timestamp, Partition, Offset, Key, and Value. The messages are as follows:

| Timestamp                | Partition | Offset | Key     | Value          |
|--------------------------|-----------|--------|---------|----------------|
| 2025-06-03T15:08:18.325Z | 0         | 0      | jóres   | Camisa do Galo |
| 2025-06-03T15:08:18.325Z | 0         | 1      | jóres   | Mouse          |
| 2025-06-03T15:08:18.325Z | 0         | 2      | jóres   | Teclado        |
| 2025-06-03T15:08:18.325Z | 2         | 0      | psliva  | Teclado        |
| 2025-06-03T15:08:18.325Z | 2         | 1      | Demando | Teclado        |

# Confluent Kafka Plataforma

## Mensagem de retorno para erro (Callback message)

8. A função `delivery_callback` tem o objetivo de enviar uma mensagem para o sistemas 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.



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

**Edite o arquivo  
CONSUMER.PY**

# Confluent Kafka Plataforma

```
consumer.py •
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 recuperação
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: %s".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 {topic}: chave = {key:12} Valor = {value:12}".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

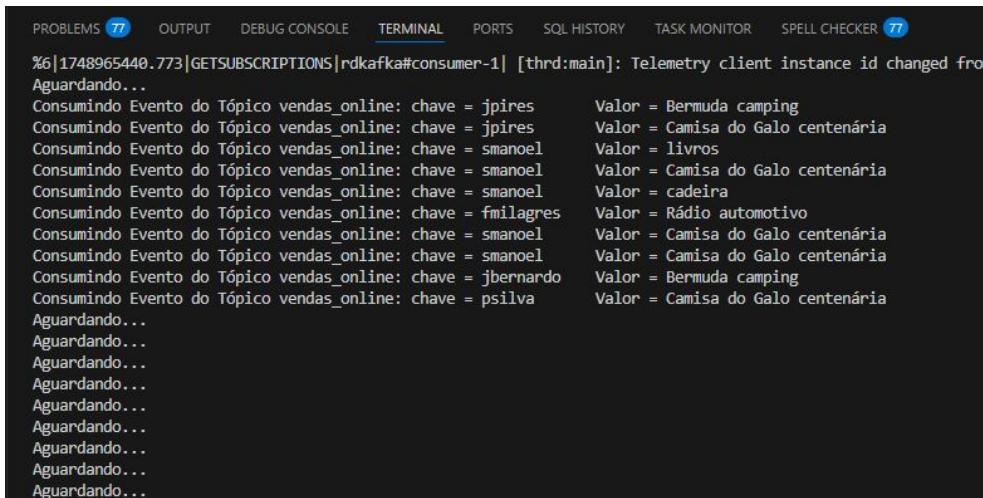
## Definindo servidor, usuário e senha

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.



```
PROBLEMS 77 OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR SPELL CHECKER 77
%6|1748965440.773|GETSUBSCRIPTIONS|rdkafka#consumer-1| [thrd:main]: Telemetry client instance id changed from
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...
```

---

## 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 Cezar. Aula 10. ICMC, 2021. Disponível em: [https://edisciplinas.usp.br/pluginfile.php/7749153/mod\\_page/content/17/Aula-10.zip](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-diferencas-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.