Laboratório Prático: Cliente Kafka em Python

Tempo estimado necessário: **30** minutos

Objetivos

Após concluir este laboratório, você será capaz de:

- Usar `kafka-python` para interagir com o servidor Kafka em Python
- Enviar e receber mensagens através do cliente Kafka-python

Sobre o Skills Network Cloud IDE

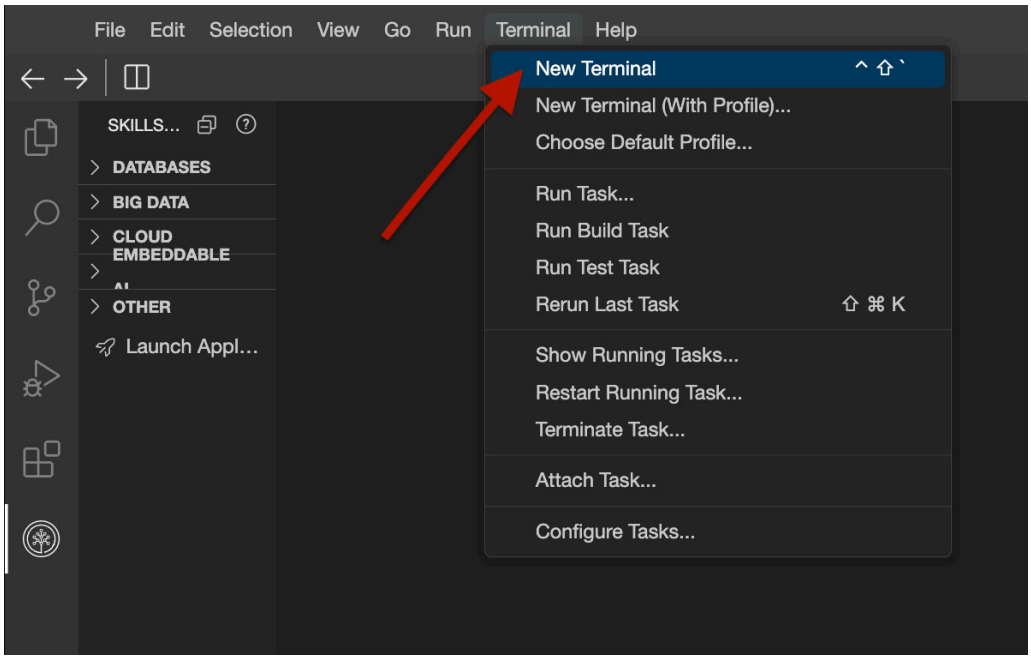
O Skills Network Cloud IDE (baseado no Theia e Docker) fornece um ambiente para laboratórios práticos relacionados a cursos e projetos. O Theia é um IDE (Ambiente de Desenvolvimento Integrado) de código aberto que pode ser executado no desktop ou na nuvem. Para completar este laboratório, usaremos o Cloud IDE baseado no Theia rodando em um contêiner Docker.

Aviso importante sobre este ambiente de laboratório

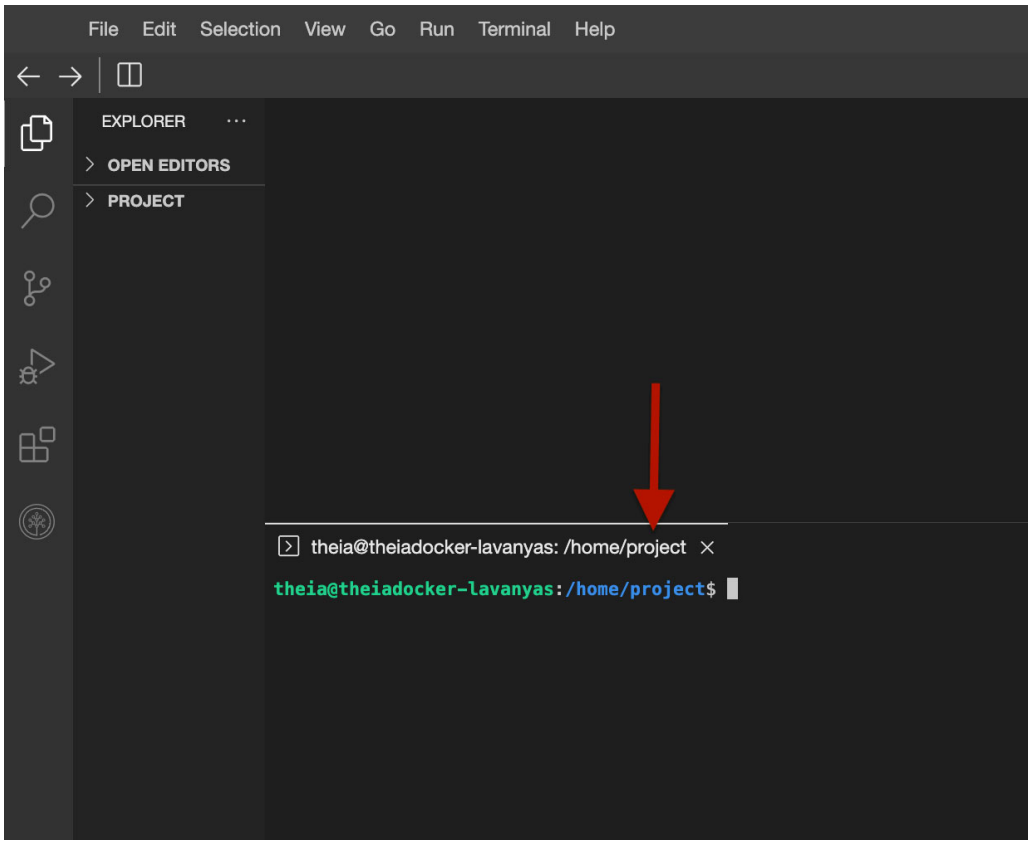
Por favor, esteja ciente de que as sessões para este ambiente de laboratório não são persistentes. Um novo ambiente é criado para você toda vez que você se conecta a este laboratório. Qualquer dado que você possa ter salvo em uma sessão anterior será perdido. Para evitar a perda de seus dados, planeje completar esses laboratórios em uma única sessão.

Exercício 1: Baixar e extrair Kafka

1. Abra um novo terminal clicando na barra de menu e selecionando **Terminal**->**Novo Terminal**.



Isso abrirá um novo terminal na parte inferior da tela.



Execute os comandos abaixo no terminal recém-aberto.

Nota: Você pode copiar o código clicando no pequeno botão de copiar no canto inferior direito do trecho de código abaixo e, em seguida, colá-lo onde desejar.

2. Baixe o Kafka executando o comando abaixo:

```
wget https://downloads.apache.org/kafka/3.8.0/kafka_2.13-3.8.0.tgz
```

3. Extraia o Kafka do arquivo zip executando o comando abaixo.

```
tar -xzf kafka_2.13-3.8.0.tgz
```

Isso cria um novo diretório `kafka_2.13-3.8.0` no diretório atual.

Exercício 2: Configurar KRaft e iniciar o servidor

1. Mude para o diretório `kafka_2.13-3.8.0`.

```
cd kafka_2.13-3.8.0
```

2. Gere um UUID de Cluster que identificará exclusivamente o cluster Kafka.

```
KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"
```

Nota: Este ID de cluster será utilizado pelo controlador KRaft.

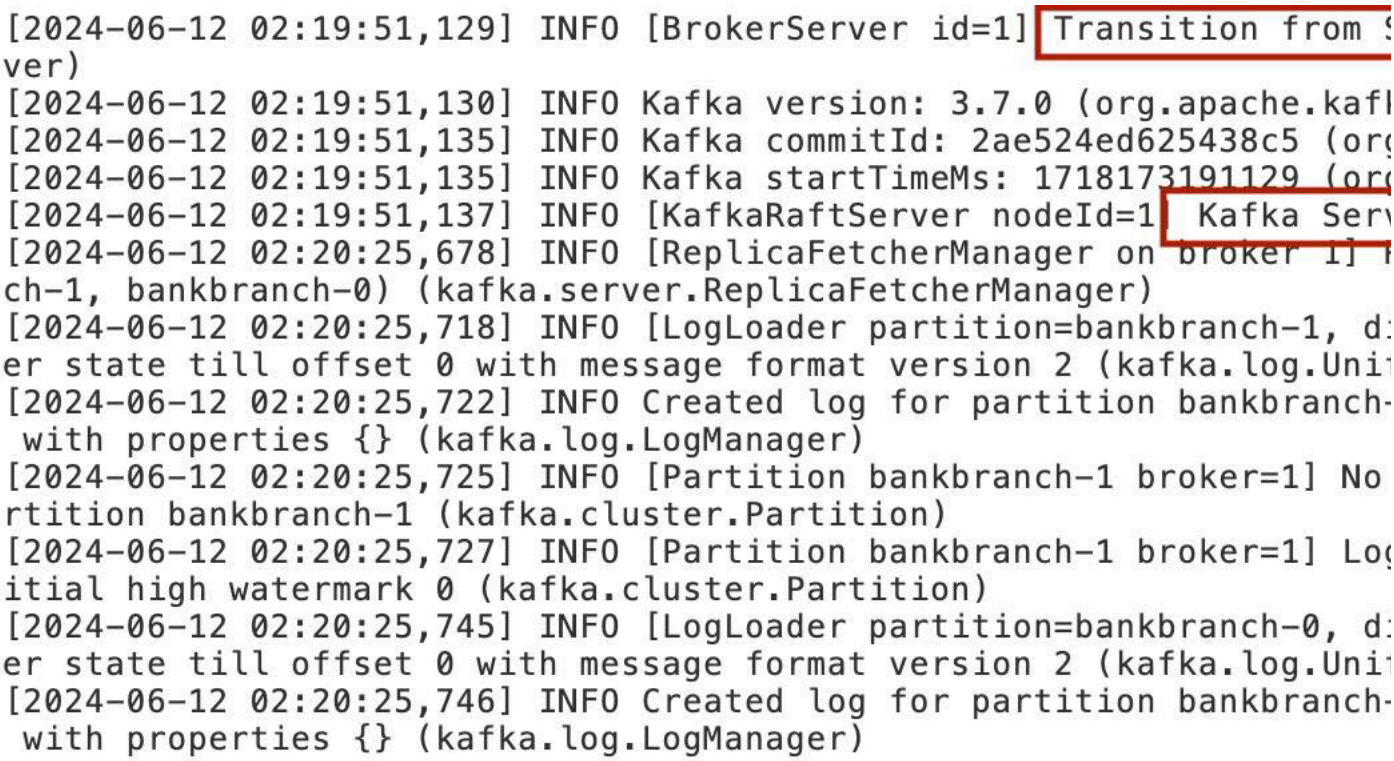
3. O KRaft requer que os diretórios de log sejam configurados. Execute o seguinte comando para configurar os diretórios de log passando o ID do cluster.

```
bin/kafka-storage.sh format -t $KAFKA_CLUSTER_ID -c config/kraft/server.properties
```

4. Agora que o KRaft está configurado, você pode iniciar o servidor Kafka executando o seguinte comando.

```
bin/kafka-server-start.sh config/kraft/server.properties
```

Nota: Você pode ter certeza de que começou quando você vê uma saída que contém mensagens que confirmam que o servidor Kafka foi iniciado com sucesso.



Exercício 3: Criar um tópico no arquivo admin.py

Instalar kafka-python

1. Abra um novo terminal e navegue até o diretório `kafka_2.13-3.8.0`.

```
cd kafka_2.13-3.8.0
```

2. Instale o pacote `kafka-python` executando o seguinte comando.

```
pip3 install kafka-python
```

3. Crie um arquivo chamado `admin.py` executando o seguinte comando.

```
touch admin.py
```

4. Clique no botão abaixo para abrir o arquivo em modo de edição, cole o seguinte conteúdo no arquivo e salve-o.

Open **admin.py** in IDE

```
from kafka.admin import KafkaAdminClient,NewTopic
admin_client = KafkaAdminClient(bootstrap_servers="localhost:9092", client_id='test')
topic_list = []
new_topic = NewTopic(name="bankbranch", num_partitions= 2, replication_factor=1)
topic_list.append(new_topic)
admin_client.create_topics(new_topics=topic_list)
```

Nota: Estamos criando um tópico "bankbranch" através deste código.

Exercício 4: Crie o arquivo producer.py

Você precisa de um produtor para enviar mensagens para o Kafka. Você encontrará o código do produtor no arquivo producer.py.

1. Crie um arquivo chamado `producer.py` executando o seguinte comando.

```
touch producer.py
```

2. Clique no botão abaixo para abrir o arquivo em modo de edição, cole o seguinte conteúdo no arquivo e salve-o.

Open **producer.py** in IDE

```
from kafka import KafkaProducer
import json
producer = KafkaProducer(value_serializer=lambda v: json.dumps(v).encode('utf-8'))
producer.send("bankbranch", {'atmid':1, 'transid':100})
producer.send("bankbranch", {'atmid':2, 'transid':101})
producer.flush()
producer.close()
```

No código acima, o produtor está enviando duas mensagens através deste código. Essas mensagens serão recebidas pelo consumidor.

Exercício 5: Crie o arquivo consumer.py

Você precisa de um consumidor para ler mensagens do Kafka. O código para o consumidor será escrito no arquivo `consumer.py`.

1. Crie um arquivo chamado `consumer.py` executando o seguinte comando.

```
touch consumer.py
```

2. Clique no botão abaixo para abrir o arquivo em modo de edição, cole o seguinte conteúdo no arquivo e salve-o.

Open **consumer.py** in IDE

```
from kafka import KafkaConsumer
consumer = KafkaConsumer('bankbranch',
                         group_id=None,
                          bootstrap_servers=['localhost:9092'],
                          auto_offset_reset = 'earliest')
print("Hello")
print(consumer)
for msg in consumer:
    print(msg.value.decode("utf-8"))
```

Exercício 6: Execute os três arquivos Python

1. Execute `admin.py` e `producer.py` usando os seguintes comandos no terminal:

```
python3 admin.py
python3 producer.py
```

2. Abra um novo terminal e execute os seguintes comandos para rodar consumer.py:

```
cd kafka_2.13-3.8.0
python3 consumer.py
```

3. Seu consumidor deve imprimir as mensagens enviadas pelo produtor da seguinte forma:

```
theia@theiadocker-lavanyas:/home/project$ python3 consumer.py
Hello
<kafka.consumer.group.KafkaConsumer object at 0x7f4e8be6bc10>
{"atmid": 1, "transid": 100}
{"atmid": 2, "transid": 101}
```

Exercício Prático

1. Crie um novo produtor a partir de `bankbranch` em um arquivo chamado `new_producer.py` que receberá a entrada do usuário enquanto o usuário desejar e aceitará a entrada do usuário para o número do caixa eletrônico com o qual deseja realizar a transação (1 ou 2) e transmitirá a transação.

2. Observe o consumidor recebendo os eventos transmitidos pelo produtor em tempo real.

▶ Clique aqui para a **solução**.

Autores

Ramesh Sana Reddy
Lavanya T S
Shreya Khurana