

My First Hands on with Apache Kafka using Python

Terms:

Zookeeper, Server, Producer, Consumer

Keypoints:

>Zookeeper stores snapshots of logs in these folders kafka-logs--zookeeper and kafka-logs--server-logs

>Kafka-broker == kafka-server

> maxClientcnxn=1 because we want only 1 server not multiple or else we will face delay

>Producer continuously keeps sending data to consumer. For consumer to process it efficiently, sleep(5) will halt next data by 5s.

>When zookeeper or kafka-server runs, many logs are generated, those logs are kept in the folder kafka-logs--zookeeper and kafka-logs--server-logs

>Kafka cluster stores messages even if they were already consumed by one of the consumers.

>Same messages may be read multiple times by different consumers.

>If a Consumer is loaded after previous-Producer&Consumer had their data sharing, even the history of that data will be reflected on the new Consumer.

>If a Producer is terminated, nothing changes, other Producers can still send messages/data.

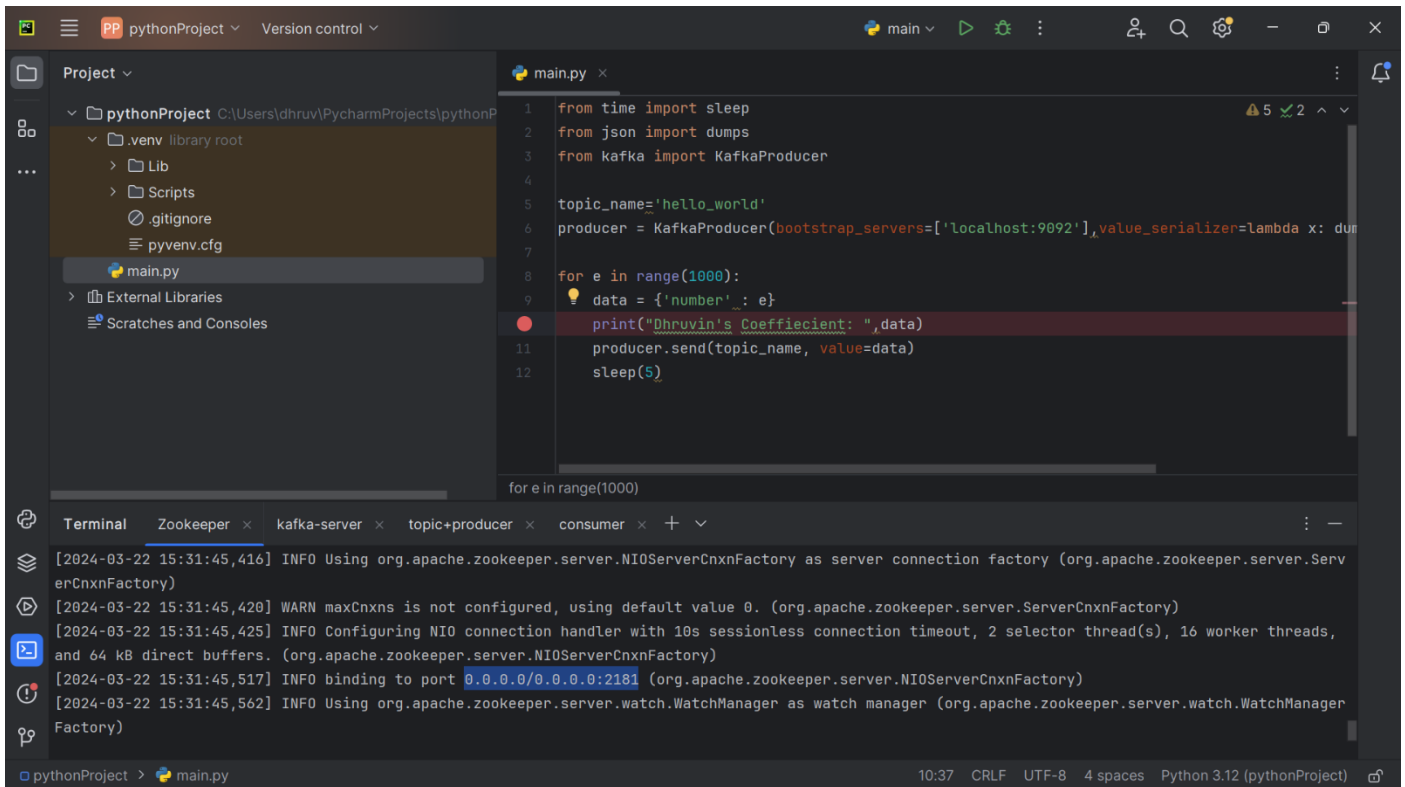
>Multiple consumers can read message from Kafka Topic in **Parallel**.

>Producers and consumers don't know about each other, they are just doing their own job, first producer do not know anything about the second producer or other consumers. Same for other components.

>Consumers do not care about the producers which has produced messages, they only consume messages.

Screenshots from my project:

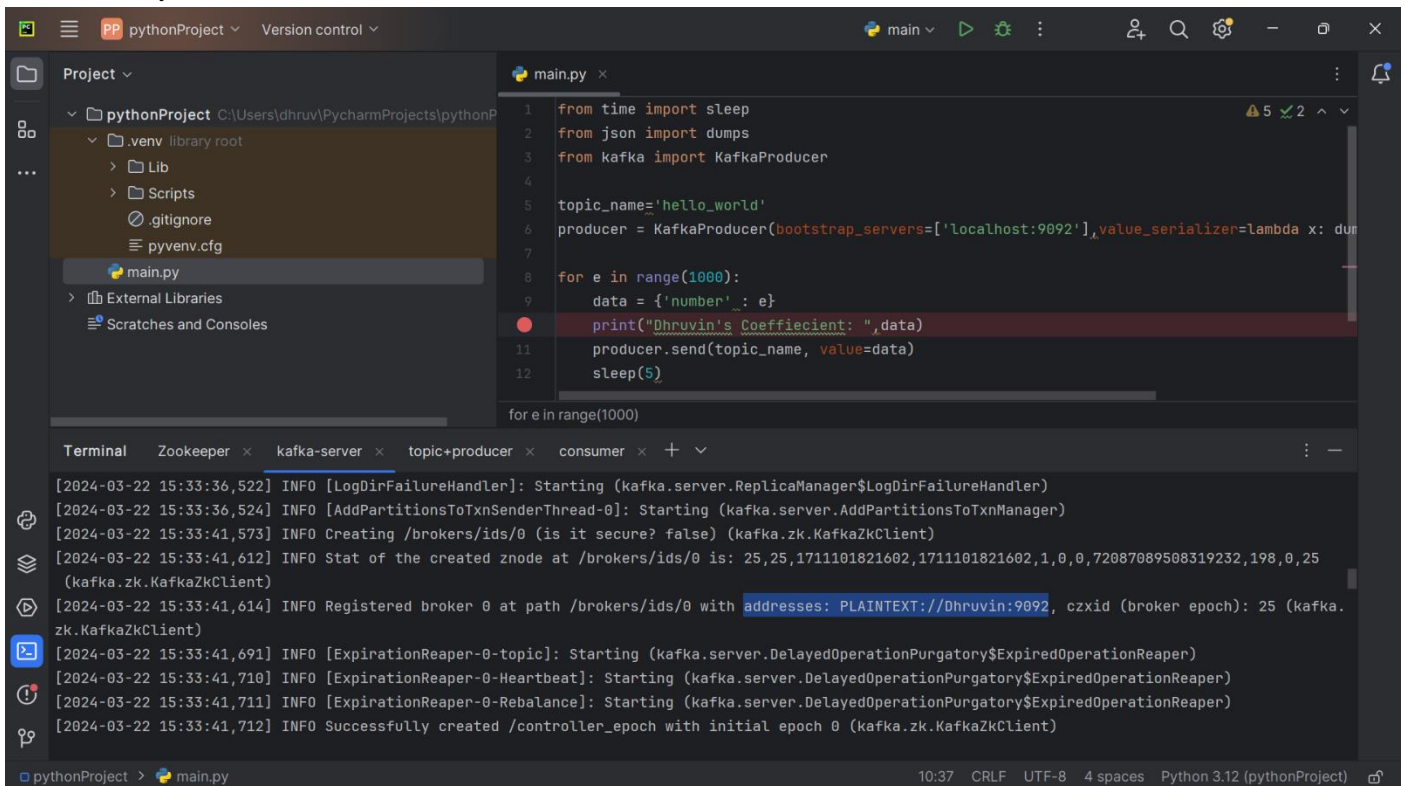
Zookeeper



The screenshot shows the PyCharm IDE with a Python project named 'pythonProject'. The 'main.py' file is open, displaying a Kafka producer script. The script imports 'sleep' from 'time', 'dumps' from 'json', and 'KafkaProducer' from 'kafka'. It sets 'topic_name' to 'hello_world' and creates a 'producer' with 'bootstrap_servers' set to 'localhost:9092'. A loop runs 1000 times, generating data and sending it to the producer, with a 5-second sleep between sends. The terminal window shows the following logs:

```
[2024-03-22 15:31:45,416] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2024-03-22 15:31:45,420] WARN maxCnxns is not configured, using default value 0. (org.apache.zookeeper.server.ServerCnxnFactory)
[2024-03-22 15:31:45,425] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 2 selector thread(s), 16 worker threads, and 64 kB direct buffers. (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2024-03-22 15:31:45,517] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2024-03-22 15:31:45,562] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
```

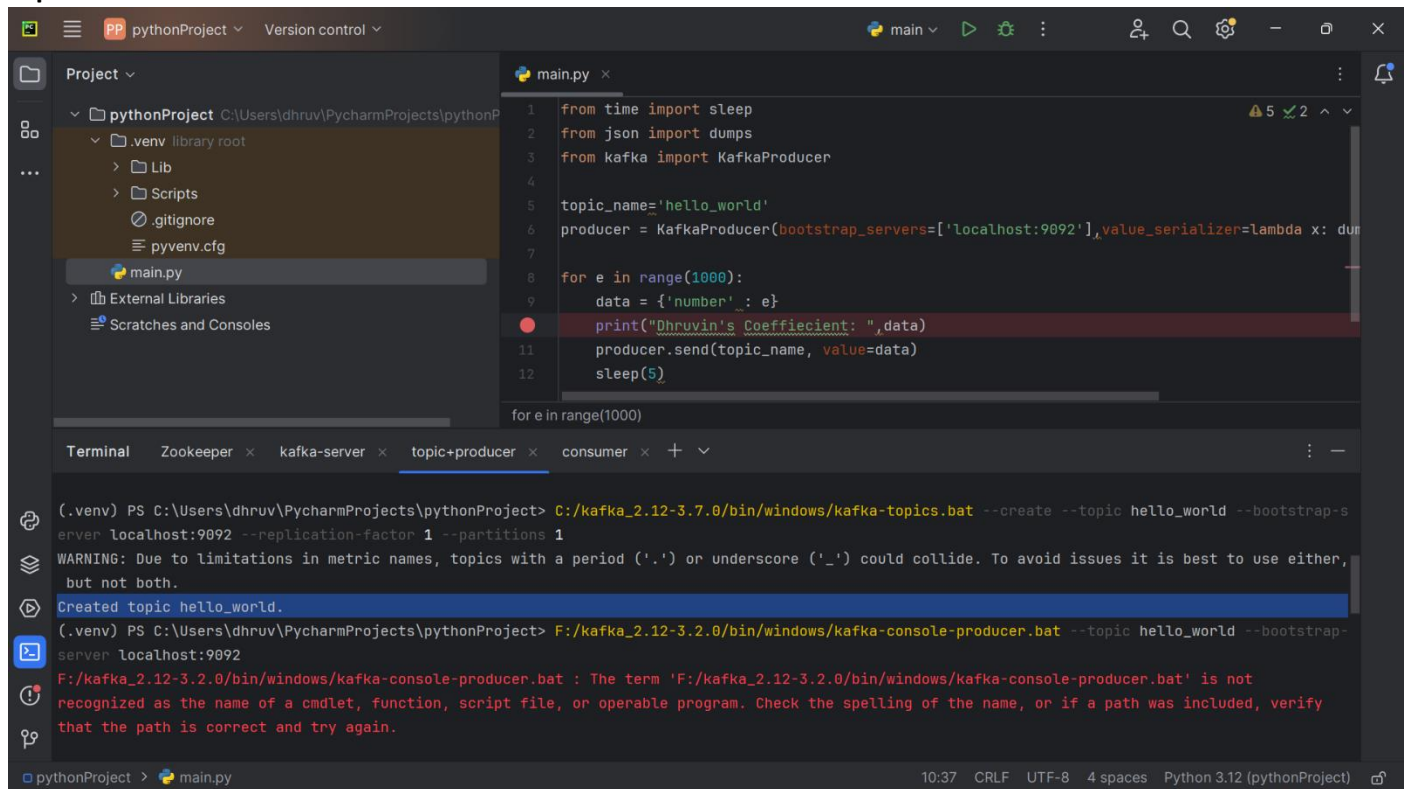
Kafka-Server/Broker



The screenshot shows the PyCharm IDE with the same 'pythonProject' and 'main.py' file. The terminal window now shows the logs for the 'kafka-server' process:

```
[2024-03-22 15:33:36,522] INFO [LogDirFailureHandler]: Starting (kafka.server.ReplicaManager$LogDirFailureHandler)
[2024-03-22 15:33:36,524] INFO [AddPartitionsToTxnSenderThread-0]: Starting (kafka.server.AddPartitionsToTxnManager)
[2024-03-22 15:33:41,573] INFO Creating /brokers/ids/0 (is it secure? false) (kafka.zk.KafkaZkClient)
[2024-03-22 15:33:41,612] INFO Stat of the created znode at /brokers/ids/0 is: 25,25,1711101821602,1711101821602,1,0,0,72087089508319232,198,0,25 (kafka.zk.KafkaZkClient)
[2024-03-22 15:33:41,614] INFO Registered broker 0 at path /brokers/ids/0 with addresses: PLAINTEXT://Dhruvin:9092, czxid (broker epoch): 25 (kafka.zk.KafkaZkClient)
[2024-03-22 15:33:41,691] INFO [ExpirationReaper-0-topic]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2024-03-22 15:33:41,710] INFO [ExpirationReaper-0-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2024-03-22 15:33:41,711] INFO [ExpirationReaper-0-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2024-03-22 15:33:41,712] INFO Successfully created /controller_epoch with initial epoch 0 (kafka.zk.KafkaZkClient)
```

Topic+ Producer



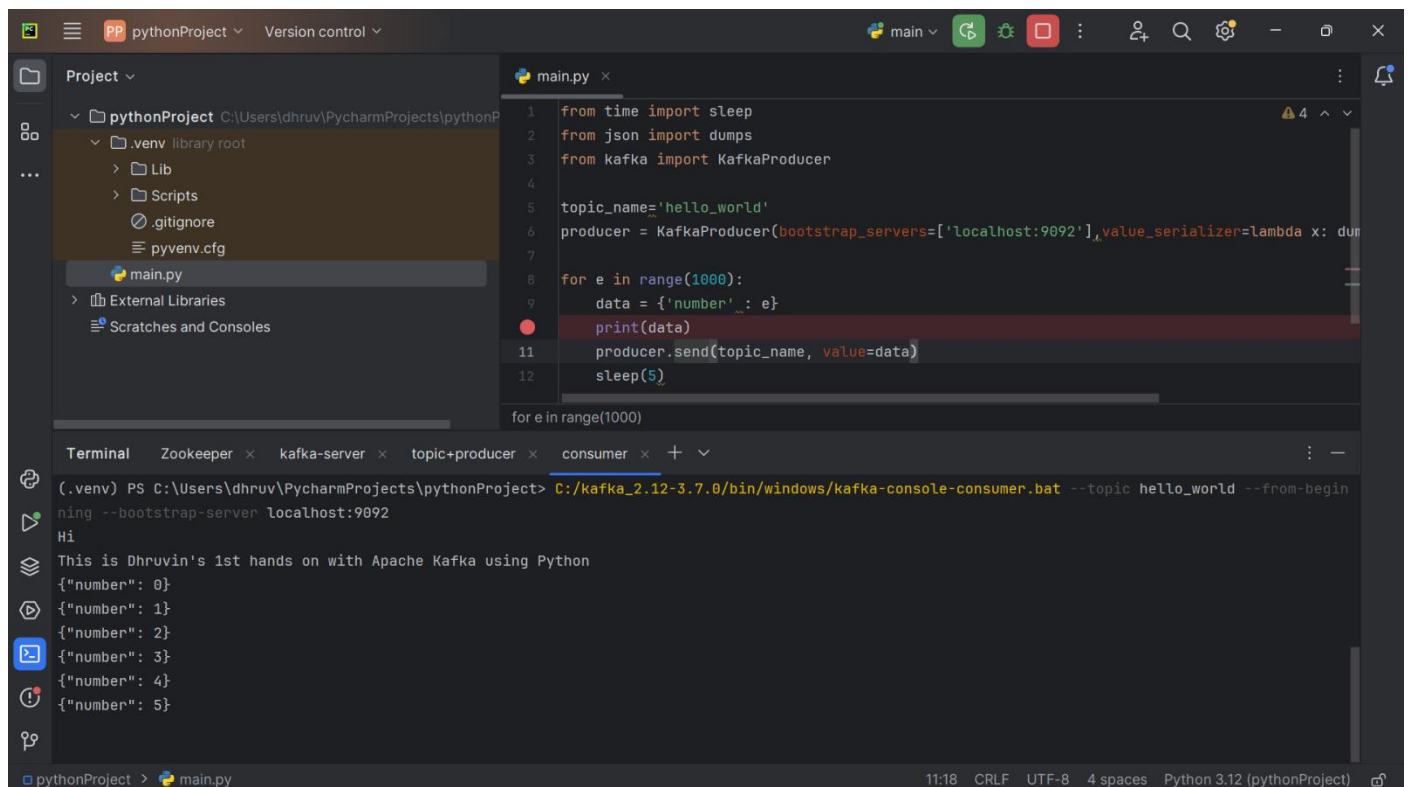
The screenshot shows the PyCharm IDE with the 'pythonProject' open. The 'main.py' file is open, showing the following code:

```
1 from time import sleep
2 from json import dumps
3 from kafka import KafkaProducer
4
5 topic_name='hello_world'
6 producer = KafkaProducer(bootstrap_servers=['localhost:9092'],value_serializer=lambda x: dumps(x))
7
8 for e in range(1000):
9     data = {'number': e}
10    print("Dhruvin's Coefficient: ",data)
11    producer.send(topic_name, value=data)
12    sleep(5)
```

The terminal shows the execution of the following commands:

```
(.env) PS C:\Users\dhruv\PycharmProjects\pythonProject> C:/kafka_2.12-3.7.0/bin/windows/kafka-topics.bat --create --topic hello_world --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic hello_world.
(.env) PS C:\Users\dhruv\PycharmProjects\pythonProject> F:/kafka_2.12-3.2.0/bin/windows/kafka-console-producer.bat --topic hello_world --bootstrap-server localhost:9092
F:/kafka_2.12-3.2.0/bin/windows/kafka-console-producer.bat : The term 'F:/kafka_2.12-3.2.0/bin/windows/kafka-console-producer.bat' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
```

Consumer



The screenshot shows the PyCharm IDE with the 'pythonProject' open. The 'main.py' file is open, showing the following code:

```
1 from time import sleep
2 from json import dumps
3 from kafka import KafkaProducer
4
5 topic_name='hello_world'
6 producer = KafkaProducer(bootstrap_servers=['localhost:9092'],value_serializer=lambda x: dumps(x))
7
8 for e in range(1000):
9     data = {'number': e}
10    print(data)
11    producer.send(topic_name, value=data)
12    sleep(5)
```

The terminal shows the execution of the following command:

```
(.env) PS C:\Users\dhruv\PycharmProjects\pythonProject> C:/kafka_2.12-3.7.0/bin/windows/kafka-console-consumer.bat --topic hello_world --from-beginning --bootstrap-server localhost:9092
```

The output of the consumer is shown in the terminal:

```
Hi
This is Dhruvin's 1st hands on with Apache Kafka using Python
{"number": 0}
{"number": 1}
{"number": 2}
{"number": 3}
{"number": 4}
{"number": 5}
```

Multiple Producers and Consumers

The screenshot shows a PyCharm IDE with a terminal window and two Command Prompt windows. The terminal window shows the creation of a Kafka topic named 'demo_testing1' and the execution of a Kafka console producer. The Command Prompt windows show the execution of a Kafka console consumer, which receives messages from the producer. The messages are: 'Hello', 'Hi this is from the single producer', 'School', 'India', 'Nepal', 'I love my Country', and 'Demo from 1st one'. The consumer also receives 'Demo from the second producer'.

```
Created topic demo_testing1.

(venv) C:\Users\USER\PycharmProjects\kafka_python>F:/kafka_2.12-3.2.0/bin/windows/kafka-console-producer.bat --topic demo_testing1 --bootstrap-server localhost:9092
>Hello
>Hi this is from the single producer
>School
>India
>Nepal
>I love my Country
>Demo from 1st one
>
```

```
C:\Users\USER>F:/kafka_2.12-3.2.0/bin/windows/kafka-console-consumer.bat --topic demo_testing1 --bootstrap-server localhost:9092
Hello
Hi this is from the single producer
School
India
Nepal
I love my Country
Demo from 1st one
Demo from the second producer
```

```
C:\Users\USER>F:/kafka_2.12-3.2.0/bin/windows/kafka-console-producer.bat --topic demo_testing1 --bootstrap-server localhost:9092
>Demo from the second producer
>
```

Try publishing messages from diff producers & observe the outcome.