

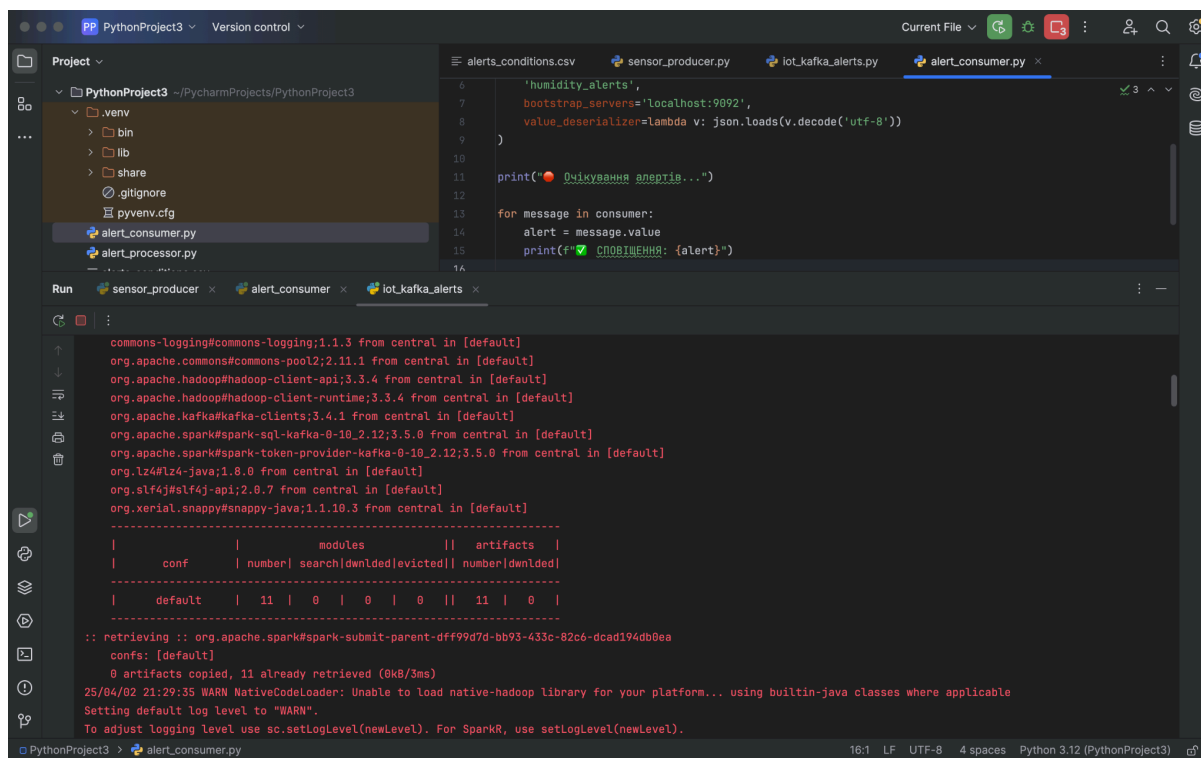
The screenshot shows the PyCharm IDE interface for a project named 'PythonProject3'. The 'Project' view on the left lists files: .gitignore, pyenv.cfg, alert\_consumer.py, alert\_processor.py, alerts\_conditions.csv, iot\_kafka\_alerts.py, and sensor\_producer.py. The 'Run' console at the bottom shows the output of the 'alert\_consumer.py' process. It displays a series of log messages for 'Очікування alertів...' followed by three successful message receipts (green checkmarks) for sensor\_id 1053. Each message contains a timestamp, temperature, humidity, and a message in Ukrainian. The messages are:

- SPОВІЩЕННЯ: {'sensor\_id': 1053, 'timestamp': '2025-04-02T21:30:43.744147', 'humidity': 81.37, 'message': 'Вологість поза межами!'}
- SPОВІЩЕННЯ: {'sensor\_id': 1053, 'timestamp': '2025-04-02T21:30:53.765211', 'temperature': 44.35, 'message': 'Висока температура!'}
- SPОВІЩЕННЯ: {'sensor\_id': 1053, 'timestamp': '2025-04-02T21:31:01.778779', 'humidity': 18.26, 'message': 'Вологість поза межами!'}

The code editor shows the 'alert\_consumer.py' file with the following code:

```
1 import json
2 from kafka import KafkaConsumer
3
4 consumer = KafkaConsumer(
5     topics=['temperature_alerts',
6            'humidity_alerts'],
7     bootstrap_servers='localhost:9092',
8     value_deserializer=lambda v: json.loads(v.decode('utf-8'))
9 )
10
```

**Робота двох процесів** одночасно. Видно кілька повідомлень з однаковим `sensor_id=1053`, але з різними значеннями `temperature` та `humidity`, що свідчить про активну генерацію даних сенсорів у Kafka-топик `building_sensors`.



iot\_kafka\_alerts.py читає дані з building\_sensors, обробляє їх за допомогою Spark Structured Streaming, фільтрує алерти на основі умов з alerts\_conditions.csv, і відправляє у Kafka-топіки temperature\_alerts та humidity\_alerts.