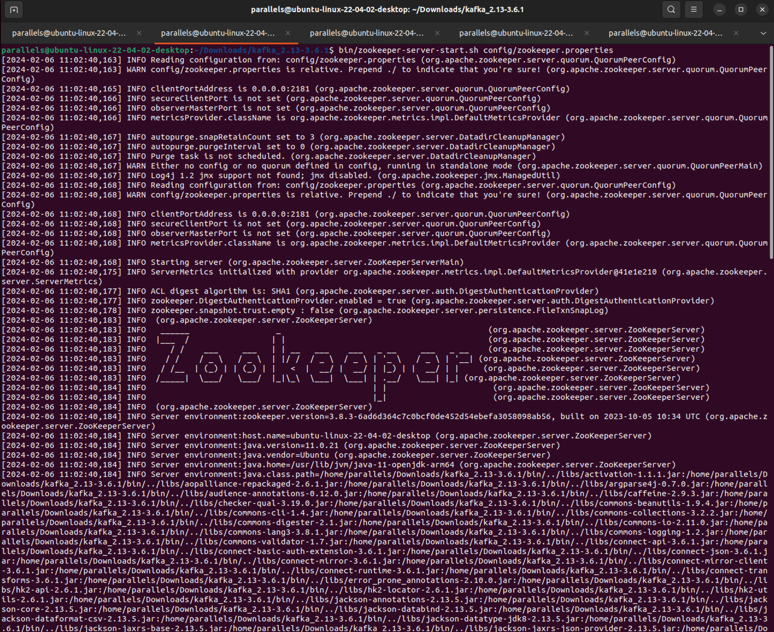
**Executive Summary – Data Streaming Process**

This executive summary explains the creation and execution of a weather data streaming system. The project's focus is to create a dynamic data pipeline by integrating Kafka with PostgreSQL. This will be achieved by using Python to simulate the generation and handling of weather metrics for two Canadian cities “Winnipeg” and “Vancouver”. The process involves setting up Kafka topics and meticulously inserting data into a robust database. The following sections will walk through each step of the process, culminating in a fully functional real-time data streaming application.

1. **Kafka Initialization:** Initiate Kafka and Zookeeper services, ensuring operational status on localhost for message brokering.

A screenshot of a computer screen

Description automatically generated



1. **Python Producer Deployment:** Crafted producer.py to simulate weather data generation, periodically publishing this data to the Kafka topic, as verified by terminal logs. The data encompass random temperature, wind speed, and humidity values within specified ranges for each city, serialized into JSON format.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Python Consumer Implementation**: Implemented **consumer.py** to subscribe and retrieve messages from the Kafka topic, prepped for database insertion. **Database Connectivity**: Established a PostgreSQL connection, confirming accessibility and readiness for data manipulation.

A screenshot of a computer

Description automatically generated

1. **Table Setup**: Executed SQL commands to generate a **weather** table, structuring it to receive weather data inputs.

A screenshot of a computer

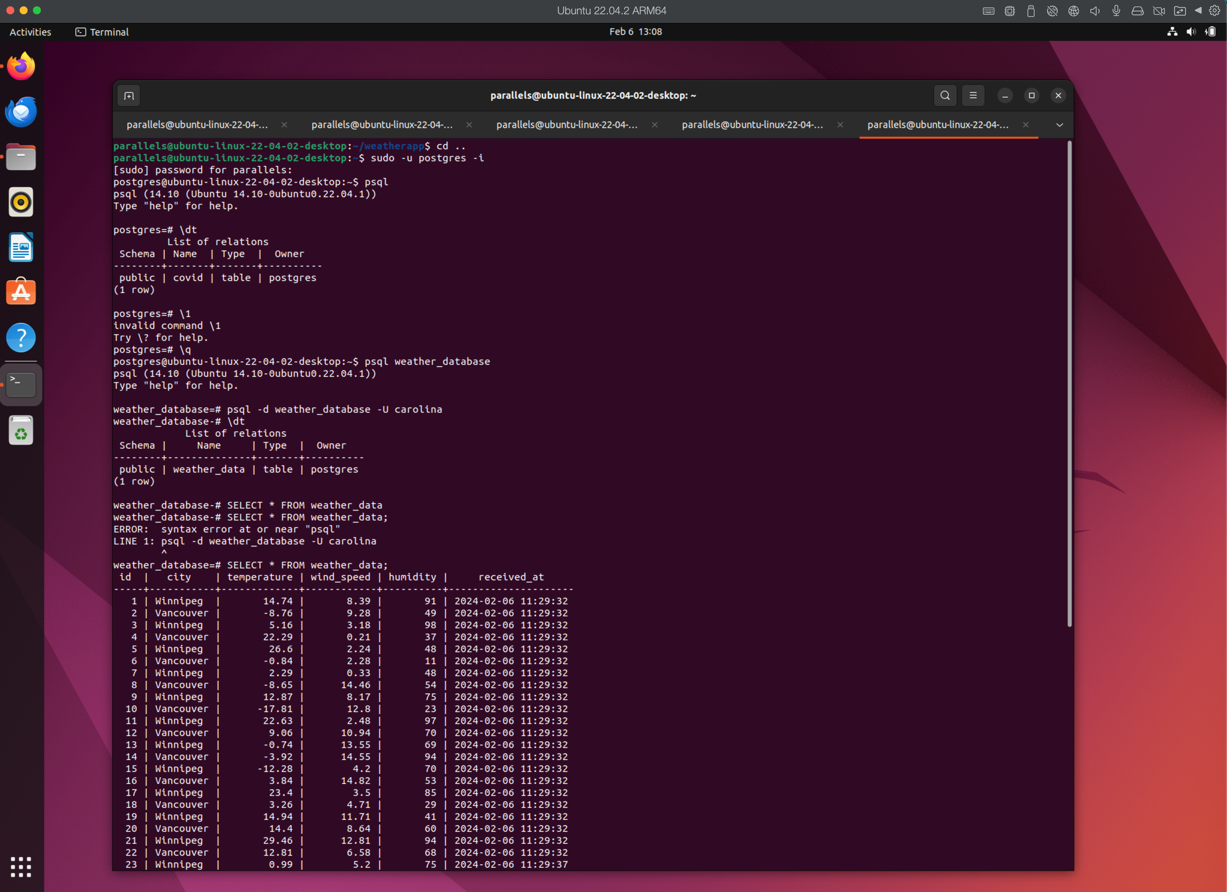
Description automatically generated

1. **Topic Management:** Utilized kafka-topics. sh to successfully create a Kafka topic, weather\_data\_topic, facilitating a channel for data streaming and executing both producer and consumer scripts in separate terminals to simulate the real-time data flow.

A screenshot of a computer screen

Description automatically generated

1. After a sufficient runtime verify the successful insertion of data into the PostgreSQL database and pgadmin4.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The project aimed to simulate and process real-time weather data for Winnipeg and Vancouver using Kafka and PostgreSQL successfully. The Python scripts, producer.py, and consumer.py were able to effectively interact with the Kafka streaming platform. They generated synthetic weather data and consumed it respectively. The PostgreSQL database reliably stored the incoming data, which was confirmed through the database interface. The pipeline ran seamlessly from data generation to storage, demonstrating the robustness of the integration. This paves the way for potential real-time data analytics and monitoring applications.