Smart Home Assistant Project

For Skills Bootcamp: DevOps/Cloud Engineering - Cloud Engineering

Context

You have been tasked with developing a system of containers utilising Event Driven Architecture to create a Smart Home Assistant capable of monitoring and controlling various IoT devices in a home environment.

The devices will collect data and trigger actions based on user-defined thresholds or events.

High-level desired outcome

As a proof of concept, you will create an Internet of Things platform that allows users to register their IoT devices to interact with each other seamlessly.

For example:

Registering a light bulb and an energy tracker device to a system responsible for managing multiple devices.

If the energy tracker detects that the light has been left on for too long and consuming too much energy, it will trigger the light bulb to turn off via the system.

This communication should occur through an Event Bus, enabling flexibility for multiple systems and devices to interact without direct coupling.

Minimum Viable Product

Your project should consist of an HTTP server with the following features:

1. An Event Bus enabling users to publish events, subscribe to events, and notify subscribers of published events.
2. A database to store information about connected components (e.g., light bulb, energy tracker) and their current states ("off," "active," "triggered").
3. Two external components that can be connected and disconnected from the system. One component should publish an event, and the other should consume it via the Event Bus as proof of concept.

Non Functional Requirements

- Implementation can be in any programming language of your choice.
- The system must be resilient to failures and designed for scalability to handle increasing device connections.
- Implementation should include monitoring capabilities to enable users to perform system audits effectively.
- The code should be well tested.

Possible Extensions

- Development of a user-friendly frontend interface for device management and monitoring.
- Creation of Helm Charts for easy deployment of the service and components to a Kubernetes Cluster if desired.
- Implementation of health check mechanisms to monitor the status of external components and ensure system reliability.

Due Date

To be advised, but not later than six weeks from commencement.