# Edge–Edge Cloud Setup: System Overview and Restart Guide

## 1. What You Have Built

You have created a complete mini edge–cloud computing system using your Raspberry Pi as the edge node and your MacBook as the edge cloud. Here’s how each component works together:

🍓 Raspberry Pi (Edge Device):  
- Generates data (currently random numbers; later, can send camera data).  
- Sends this data via MQTT to your MacBook.  
- Acts as a simple edge node.

💻 MacBook (Edge Cloud):  
- Runs Mosquitto MQTT broker to receive data from the Pi.  
- Runs Docker, which is the container runtime.  
- Runs K3d, a lightweight Kubernetes cluster inside Docker.  
- Runs KubeEdge CloudCore, the orchestration layer that would manage multiple edge nodes.  
- Hosts your subscriber container inside Kubernetes, which subscribes to MQTT messages and processes them.

Together, these form a local edge–cloud pipeline:  
  
Pi → MQTT (Mosquitto on Mac) → K3d (Kubernetes) → Containerized Subscriber → Output & Analysis  
  
This setup mimics a real-world industrial edge system with a local ‘cloud’ node and an edge device.

## 2. Relationship Between Docker and K3d

Docker provides the foundation — it runs the containers. K3d uses Docker to run a lightweight Kubernetes cluster (K3s) inside containers. Kubernetes (via K3d) then orchestrates your workloads (like your MQTT subscriber container).

In short:  
- Docker = Container engine.  
- K3d = Runs a Kubernetes cluster inside Docker.  
- Kubernetes = Manages which containers run and where.  
- Your subscriber container = One of those managed workloads.

## 3. Restart Guide (Next Day Setup)

If you shut down your MacBook and Raspberry Pi, here’s how to start everything again the next day.

### A. On the Raspberry Pi:

1. Power on the Raspberry Pi and open the terminal.

2. Navigate to your project folder:

cd ~/edge\_publisher

3. Start the publisher script:

python3 pi\_publisher.py

(This will start generating and publishing data again.)

### B. On your MacBook (Edge Cloud):

1. Start Docker Desktop if it’s not running.

2. Start Mosquitto MQTT broker:

brew services start mosquitto

3. Start your K3d Kubernetes cluster:

k3d cluster start edgecloud

4. Check that the cluster is running:

kubectl get nodes

5. Check that your MQTT subscriber pod is running:

kubectl get pods

If it’s not running, redeploy it with:

kubectl apply -f subscriber-deployment.yaml

6. Check the logs to verify data is being received:

kubectl logs -f deployment/mqtt-subscriber

If everything is working, you’ll again see live data arriving from the Raspberry Pi in your Kubernetes-managed subscriber container.

## 4. Optional Verification Steps

- To test MQTT manually: open a terminal and run `mosquitto\_sub -h localhost -p 1884 -t 'sensor/data'`.  
- To view all running containers: `docker ps`.  
- To stop the cluster cleanly: `k3d cluster stop edgecloud`.  
- To restart CloudCore: re-run `docker start cloudcore` if it was stopped.

## 5. Summary

You now have a complete local edge–cloud computing environment where your Raspberry Pi acts as an edge sensor node and your MacBook acts as a local edge cloud orchestrating and processing data. This architecture mirrors real-world IoT and edge-cloud deployments, giving you the flexibility to extend into AI, visualization, or cloud integration next.