**Serverless Distributed Systems (623.720), Fall 2024**

**Final Project Description**

**Deadline**:

---

Each group will design and implement an **application (DAG format)** that spans cloud and edge components, deployed on Google Cloud Platform (GCP) using container-based technologies. You need to:

- Develop, debug, and test the **multi-stage** workflow locally before deploying it to GCP, ensuring it works well and then thinking about GCP deployment.
- Your project should include at least **three GCP nodes**:
    - **Edge instance**: Runs on a small GCP VM (e.g., collecting data, running simple AI model, pre-processing).
    - **Cloud instance**: Runs on a regular GCP VM (e.g., running dashboards or heavier analytics).
    - **Storge instance**: used to store or fetch intermediate results (e.g., via buckets or databases).
- Cloud and edge instances must communicate together via cloud storage buckets, a simple cloud database, **or** file transfer.
- Use **Docker** and Docker Compose to containerize each part of your application.
- Use **scripts** based on shell scripts, gcloud CLI, **or** tools like Terraform to automate the **VMs creation**, **package installation**, **firewall opening**, **container deployment** and **whatever** you need before running the application.
- Use a load testing tool like **Locust** to simulate real usage or load (e.g., many sensor messages).
- Design scenarios to test the workflow under various conditions, like **high concurrency**, simulate many concurrent application requests and analyze how the system scales.
- Configure **cloud monitoring and logging** to track and benchmark metrics, **like**:
    - **Execution time**: Measure how long each task in the workflow takes to complete
    - **Resource utilization**: Track memory and CPU usage for each task.
    - **Network latency:** Measure between edge and cloud
    - **Auto-scaling behavior**: Observe how the workflow scales up and down under varying loads.

- Based on your measurements, evaluate collected metrics and analyze:
    - **Performance**: Assess latency and throughput for the workflow under different conditions.
    - **Scalability**: Determine how the system performs under increasing loads, focusing on the number of instances spawned and their impact on performance.
    - Make sure to summarize your key findings and discuss the implications of different system configurations.
- If you are interested, you may also (**Optional**):
    - Move your app to **Kubernetes** (Minikube or GKE).
    - Create the necessary YAML files for deployment and services.

- **What to submit and present**
    - Upload a single **.zip file (groupName_FinalProject_IoT25)** to Moodle that includes:
        - All application **source code** and **Dockerfiles**
        - All **scripts**
        - A README with setup and run instructions
    - A report with the name of active members, including explanations and screenshots of:
        - A detailed **overview of the workflow design** and **stages**.
        - Machines, implementation, deployment, and preparation steps.
        - **Test scenarios** and benchmark results, supported by graphs and tables if needed.
        - Observations on performance, cost, and scalability.
        - **Recommendations** for improving cloud–edge application performance based on your findings.
    - Presentation file.

***Note that you can upload your zip file after your class presentation.***