Task 2

Steps

- **Step 1: deploy** a simple RAG pipeline
- **Step 2: measure** the deployed RAG service
- **Step 3: optimize** the system with techniques learned in this course
    - Step 3.1: Implement a request queue
    - Step 3.2: Implement a batcher

Step 1

- **Deploy** a simple RAG pipeline (`serving_rag.py`)
- Setup (link)
    - Login to the slurm cluster
    - Follow up the document to run a service (recommend: `srun`)

- Examples to use `srun`

Step 2

- **Measure** the deployed RAG service
- Requirements
  - Implement your scripts to test the deployment with different request rate
  - Measure the system throughput and latency (i.e., request completion time)
  - Report key metrics and analyse what is the system capacity? what is the current bottleneck?
  - Describe how do you test and measure the system performance
- Reference
  - https://github.com/ServerlessLLM/TraceStorm

Step 3

- **Optimize** the system with techniques learned in this course
  - Step 3.1: Implement a request queue
  - Step 3.2: Implement a batcher
- Measure and analyze the improvement of each optimization

Step 3.1

- **Implement** a request queue by modifying the provided script
- A potential design:
  - Create a request queue
  - Put incoming requests into the queue, instead of directly processing them
  - Start a background thread that listens on the request queue
- Hints:
  - Check out those "hints" in the code
  - Feel free to implement your own design!

Step 3.2

- **Implement** a batcher based on 3.1
- A potential design
    - Take up to MAX_BATCH_SIZE requests from the queue
    - Wait until MAX_WAITING_TIME if current batch size < MAX_BATCH_SIZE
- Hints:
    - It's ok to hardcode hyperparameters (such as max batch size and waiting time)