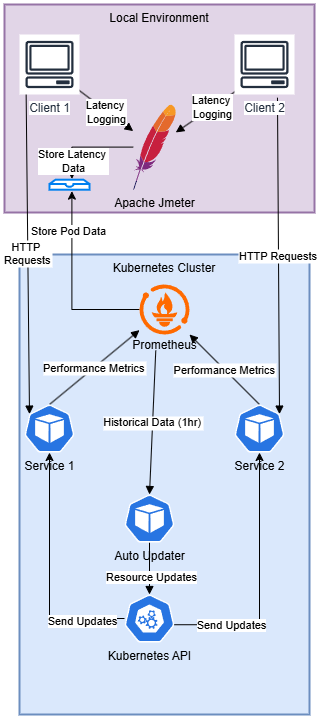
# Experimental Setup(Details under this topic are only for Service 1 & Service 2)



## Kubernetes Cluster Components:

• Go service (Service 1) implementing an echo service

• Java service (Service 2) implementing prime number verification

• Prometheus for comprehensive metrics collection

• Resource Updater for dynamic CPU & Memory allocation

## Local Environment:

• JMeter: Load Generation, precise latency measurement and data collection

## Methodology:

We have reduced CPU & Memory in equal steps in equal time differences(4 hrs) & studied the latency. When we calculating the latency, we took 100 concurrent requests, recorded latency & took the average of those latencies for plotting.

—----------------------------------------------------------------------------------------------------------------------------

# Other Services (Details under this topic are only for the Hash generator service & random password generator service)

We have implemented another 2 services to experiment with random number of requests per second using Python Scripts.

Hash generator service, Random Password Generator service services between 12 - 25 requests per second randomly.

—----------------------------------------------------------------------------------------------------------------------------

# Results

We have done 3 experiments with each service.

1. Reduction of CPU only at one time
2. Reduction of Memory only at one time
3. Reduction of both CPU & Memory at the same time

All Plots: <https://github.com/cepdnaclk/e19-4yp-AI-Dirven-Latency-Constrained-Resource-Management-In-Kubernetes/tree/main/results>