**Experiment**: - Gradual Pod Failure with Load Testing

**Description**

Chaos engineering helps test the **resilience** of applications by simulating real-world failures like **CPU stress, network latency, pod failures, and load spikes**. This experiment aims to analyse system behaviour under different failure conditions

**Experiment Workflow & Its Impact**

1. **CPU Stress** – Initially, the system undergoes high CPU usage to simulate peak load conditions.
2. **Network Latency Simulation** – Gradual increase in network delay:
   * **10% latency (2000ms) for 1 minute**
   * **20% latency for 1 minute**
   * ...
   * **100% latency for 1 minute**
3. **Pod-Kill** – The target pod is forcibly terminated to check system recovery.
4. **Load Test** – The system is stress-tested post-recovery to evaluate performance and stability

**Experiment workflow**

1. CPU stress   
2. network latency 10% for 1 min (latency: 2000ms)  
2. network latency 20% for 1 mins  
3. network latency 30% for 1 mins  
4. network latency 40% for 1 mins  
5. network latency 50% for 1 mins  
6. network latency 60% for 1 mins  
7. network latency 70% for 1 mins  
8. network latency 80% for 1 mins  
9. network latency 90% for 1 mins  
10. network latency 100% for 1 mins  
11. Pod-Kill  
12. Load test  
  
cd /root/hymavathi/

**Install locust tool, and we need to load test based out testing**

[Locust - A modern load testing framework](https://locust.io/)

**Expected output**

I am testing based on the load to check how many requests are coming in  
Identify recovery time post pod failure  
Measure response times and request success rates