📡 Continuous Monitoring

Implement continuous performance monitoring in the production environment to ensure ongoing stability, detect issues proactively, and refine performance tests as the application evolves.

**Purpose:**

* **Monitor** application performance continuously in production.
* **Review & refine** performance tests as usage patterns change.
* **Detect** performance issues before they impact users.

**🎯 Key Objectives**

1. Monitor the application’s performance in real time.
2. Detect and address performance issues proactively.
3. Maintain optimal performance under varying loads.

**🛠️ Steps for Continuous Monitoring**

**1. 🧩 Setting Up Monitoring Tools**

* Choose tools (e.g., New Relic, Dynatrace, AppDynamics) based on requirements.
* Configure monitoring for key metrics: response times, throughput, error rates, CPU, memory, disk I/O, and network usage.

**2. 📏 Defining Key Performance Indicators (KPIs)**

* Identify KPIs critical to application health.
* Examples: average & peak response time, request rate, error rate, CPU usage, memory usage, disk I/O.

**3. 📊 Establishing Baselines & Thresholds**

* Use historical or initial test data to define baselines.
* Set thresholds to trigger alerts for deviations.

**4. ⏱️ Real-time Monitoring & Alerts**

* Continuously observe performance metrics.
* Configure automated alerts for anomalies or breaches.

**5. 🔍 Analyzing Performance Data**

* Review performance data for patterns & trends.
* Use historical data for capacity planning and predictions.

**6. 🛡️ Proactive Issue Resolution**

* Investigate and resolve issues promptly.
* Collaborate with dev & ops teams to address root causes.

**7. 📝 Reporting & Documentation**

* Generate regular reports on application health.
* Record identified issues and corrective actions taken.