1. Infrastructure Metrics 🖥️

CPU Utilization: Ideal **< 70%**
PromQL:
100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100)

Load Average: Should be **lower than the number of CPU cores**
PromQL:
node_load1

CPU Steal Time: Ideal **< 2%**
PromQL:
rate(node_cpu_seconds_total{mode="steal"}[5m]) * 100

Memory Utilization: Should be **< 80%**
PromQL:
100 * (1 - (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes))

Swap Usage: Ideal **< 10%**
PromQL:
100 * (node_memory_SwapTotal_bytes - node_memory_SwapFree_bytes) / node_memory_SwapTotal_bytes

Page Faults: Should be **low & stable**
PromQL:
rate(node_vmstat_pgmajfault[5m])

Disk Usage: Should be **< 80%**
PromQL:
100 * (node_filesystem_size_bytes{fstype!="tmpfs"} - node_filesystem_avail_bytes{fstype!="tmpfs"}) / node_filesystem_size_bytes{fstype!="tmpfs"}

Disk I/O Latency: Ideal **< 10ms (SSD), < 50ms (HDD)**
PromQL:
rate(node_disk_io_time_seconds_total[5m])

Network Latency: **< 100ms (internal), < 300ms (external)**
PromQL: *(Not directly available in Prometheus, needs blackbox exporter)*

Packet Loss: Should be **0%**
PromQL: *(Use blackbox exporter for ICMP packet loss tracking)*

TCP Retransmissions: Ideal **< 1%**
PromQL:
rate(node_netstat_TcpRetransSegs[5m]) / rate(node_netstat_TcpOutSegs[5m]) * 100

---

2. Application & Service Metrics 📊

Uptime (Availability): Should be **99.9%+**
PromQL:
up

Error Rate (HTTP 5xx errors): Ideal **< 0.5%**
PromQL:
sum(rate(http_requests_total{status=~"5.."}[5m])) / sum(rate(http_requests_total[5m])) * 100

Latency (Response Time): API **< 200ms**, Web **< 1s**
PromQL:
histogram_quantile(0.95, sum(rate(http_request_duration_seconds_bucket[5m])) by (le))

Requests per Second (RPS/QPS): Should be within **normal range**
PromQL:
sum(rate(http_requests_total[5m]))

Concurrency (Active Users): Should be **within expected limits**
(No direct PromQL, depends on application metrics)

Cache Hit Ratio: Ideal **> 80%**
(Varies based on cache tool; e.g., Redis, Memcached, or CDN monitoring)

API Response Time: Should be **< 300ms**
(Same as latency query above)

DB Query Execution Time: Should be **< 100ms**
(Not directly available in Prometheus, needs database exporter like MySQL or PostgreSQL)

---

3. Logs & Error Monitoring 📜

- **Application Logs (Errors, Warnings, Info)**: Should be **low and predictable**
  (Use Loki/Grafana for log monitoring, no direct PromQL equivalent)
- **System Logs (Kernel, dmesg, etc.)**: Should have **minimal warnings**
  (Loki query required, not PromQL)
- **Exception Rate**: Should be **< 1%**
  (Use logging tools like ELK, Loki, or Datadog for error monitoring)
- **Alert Trends**: Should have a **stable trend**
  (Alerts can be monitored via Alertmanager metrics in Prometheus)

---

4. Security Metrics 🔐

- **Failed Login Attempts**: Should be **< 5 per user/day**
  (Requires security logs; use ELK/SIEM for tracking)
- **Audit & Access Logs**: Should show **normal access patterns**
  (Use SIEM tools, not Prometheus)
- **Firewall & IDS Alerts**: Minimal or no critical alerts
  (Track via IDS logs like Suricata or AWS GuardDuty)

---

5. DevOps & CI/CD Pipeline Metrics 🚀

- **Deployment Frequency**: Ideally **multiple times per day/week**
  (Track via Jenkins/GitLab, not Prometheus)

- **Change Failure Rate**: Should be **< 15%**
  (Tracked via deployment logs)
- **Mean Time to Recovery (MTTR)**: Should be **< 1 hour**
  (Can be tracked in Prometheus using alert resolution metrics)
- **Lead Time for Changes**: Ideally **< 1 day**
  (Tracked via CI/CD tools like Jenkins, GitHub Actions, GitLab CI/CD)
