

# DAY 4: INTRODUCTION TO MONITORING

## What is Monitoring

**Monitoring** helps teams understand the state of their systems based on gathered data. This stage is crucial in gathering real-time insights on:

- Performance
- Health status
- Scalability

Monitoring helps teams collect, store, and interpret application system data. A team can use this data to:

- Reduce costs
  - Improve productivity
  - Increase system reliability
- 

## Metrics

**Metrics** express a value relevant to the system at a specific point of time. Key metrics for monitoring system health and performance:

- Latency: Latency is the time between the start of an event, such as serving a request, to its completion. This is a key indicator of performance.
- Traffic/Connections: Traffic is the amount of system usage over time. The result could require scaling to maintain performance.
- Errors: An error is an invalid state our system has reached. This is things such as a memory limit or reading a corrupted data file. Multiple errors can indicate deeper issues.
- Saturation: Saturation describes the load on our system's resources. Even an approach on our system will result in a lack in performance.

Tracking these metrics can give teams a broad view of system health and help diagnose issues.

Some metrics we can retrieve include but not limited to:

- Page load time
- User logins: successes, failures, time taken, daily active users, weekly active users
- Searches: number performed, latency

- Databases: query latency, transactions per second
  - Standard OS metrics: CPU, memory, network, and disk usage.
- 

## **SLOs, SLAs, and SLIs**

This is where SLOs, SLIs, and SLAs can help tie business goals and objectives to the data he is receiving.

### **Service Level Objective (SLO)**

A **Service Level Objective** is a range of valid measurements for a metric. SLOs might be defined in terms of:

- Latency
- Availability
- Error rate
- And many more metrics!

A goal is only helpful if we know where we are in relation to it. An SLO is where we want to be.

### **Service Level Indicator (SLI)**

A **Service Level Indicator** is the current measurements of a metric related to an SLO.

### **Service Level Agreement (SLA)**

A **Service Level Agreement** is a contract with consumers, and they carry a degree of legal responsibility. SLA binds the business to business's customers.

SLOs, SLIs and SLAs allow businesses to tie promises and goals to the data coming from their systems.

Receiving alerts when performance jeopardizes the SLA is not best practice. Breaking an SLA spells trouble. Therefore, alerts should be sent sooner. Alerts should be sent when a malfunction in the system is breaking an SLO.

---

## **Monitoring Tools**

There are several monitoring tools that can aid in understanding Lenny's infrastructure. The reason why is:

- Oversee resources such as application services, databases, and other processes.
- View system health.
- Capture real-time health statuses and performance metrics.

Monitoring is achieved using a combination of tools. Big picture trends, like performance, can be monitored by a tool like Prometheus. The time taken for a database query, may be monitored using Monyog.

Some monitoring tools on the market are:

- Zabbix: Zabbix can directly monitor Java applications and offers built-in graphing and visualization capabilities.
- Prometheus: Prometheus collects metrics values from target systems. Which lets users select data in real time.
- Sensu: Sensu is a highly extensible and scalable system that monitors cloud infrastructures.
- Monyog: Monyog is used by database teams to detect issues affecting MySQL database performance.

Monitoring tools offer a clearer picture of how applications and infrastructure are working.

Tools should provide insights to:

- Application performance
  - Application health
  - Database
  - Infrastructure
- 

## **Alerting**

When a system experiences an issue and isn't able to fix itself, an alert is triggered. Usually it's a change of state. There are a variety of ways to deliver alerts:

- Tickets
- Email alerts
- Slack alerts
- SMS/phone calls

We may develop alert fatigue if we receive an alert for everything. This could create a mentality where we start ignoring alerts or turning them off.

To avoid alert fatigue:

- Only alert when immediate human intervention is required
- Alert based on customer facing issues
- Set clear ways to indicate urgency
- Ensure an alert is not a copy of another

Alerts provide context to help teams solve an issue before it becomes a crisis.

---

## Observability

Some steps a team might go through when an issue arises:

- Evaluate usage and performance data
- Identify the cause of the issue.
- Apply the appropriate solution, restoring system performance.

**Observability** is the degree to which a system's information can be used to locate and fix a problem. A system with a highly observable system can trace and diagnose a problem.

To improve a system's observability:

- Make sure team is aligned with service level objectives
- Create meaningful alerts
- Optimize application logging by ensuring messages are informational and descriptive
- Automate work processes

Maintaining an observable system enables teams to proactively monitor and track for errors. A system with a low observability will have a primary effect of increased amount of time spent on fixing bugs could surpass the amount of time spent on delivering new features.

---

## Monitoring Quality

Some of the practices to improve monitoring quality:

- Define actionable alerts that are customized to the needs of our organization.

- Collect application logs and make this data available and understandable.
- Incorporate logging into the build and deployment process.
- Define custom, actionable alerts that are relevant to the organization.

Another key indicator of monitoring quality is the way our organization handles alerts. There are some things to look out for with the monitoring quality:

- False positives: This occurs when an alert is generated, but there is nothing wrong with the system. An alert may need to be adjusted, or deleted.
- Unactionable alerts: This type of alert has little to do with a problem and doesn't need anything done.

Useless or incorrect alerts add to the chance that valuable alerts will be ignored or unseen.

The benefits of high-quality monitoring:

- Easily understanding the system status.
  - Fast discovery of issue causes.
  - Reduced employee burnout and alert fatigue.
- 

## **Review**

- Monitoring is the practice of collecting metrics to gain insights on our systems.
- Insights into our systems tell us about its health and problems.
- We can tie our business objectives to metrics with SLAs, SLOs, and SLIs.
- Alerts should be sent out for customer facing issues requiring immediate attention.
- We can use monitoring tools to aggregate and visualize our system insights.
- Observability is the degree to which our monitoring helps us solve system problems.
- A high-quality monitoring system allows us to better diagnose and resolve system issues!

Monitoring is critical to applying DevOps in any organization. Start gaining system insights with monitoring today!