**OBSERVEBILITY ZERO TO HERO**

With observability we can have the data of the servers or nodes or pods like

Disk utilization of nodes

CPU utilization

Memory utilization

No. of http requests that are successful and no. of requests that are failed.

We can find the state of the node or server which helps in understanding the applications perform.

Along with all the information of the nodes we can understand why its happening and can help in resolving the issues.

What? Is happening

Why? Is it happening

How? to fix it

All these things will be obtained by observability.

**Three pillars of Observability:**

1. **Metrics**

What is the state of the system?

Provides historical data of events as no one can predict when there will be a spike in the CPU utilization or Memory etc. Only with the metrics history we can find the data which happened in node.

1. **Logging**

Why is the system in that particular state?

To clearly understand the issue, we need exactly which request caused the spike in CPU at that particular time and which part of the application was hit with that request and caused the CPU spike.

1. **Traces**

How to fix it?

Traces will give extensive info of the request

**Example a http request failed**

Now we find all the requests that are failed in the last 30 mins. All requests that are failed in last 30 mins are found in the metrics as historical data.

The information of the request like source and destination and the reason why it got failed is available in the Logs.

The request from the source (client) will be moved from Load Balancer >> front end >> backend >> Database and it goes back. This entire path will be traced and kept in traces with which we can find the solution and fix it.

Observability covers all three pillars metrics, logs and traces.

**Monitoring:**

Metrics + Alerts + Dashboards all these will make monitoring.

If there is a CPU spike in the node and with metrics the alerts are sent via mail or notifications. Now a dashboard will help in display of the alerts.

**Service Level Agreement:**

SLA is an agreement made between the service provider and customer where the provider promises to ensure the availability of the application so that customers will not face issues.

**Metrics:**

**Prometheus:**