

Observability - Week 8

Assignment Report

Siddharth Kumar

September 24, 2025

Contents

1	Task 1: Telegraf Deployment and Monitoring	2
1.1	Telegraf Installation	2
1.2	Telegraf Configuration	2
1.3	Starting Telegraf and Verification	5
2	Task 2: Grafana Visualization	6
2.1	Grafana Setup	6
2.2	Adding OpenTSDB as a Data Source	7
2.3	Visualizing Metrics in Grafana	7
3	Task 3: Inducing and Observing Service Impact	11
3.1	Description of Induced Failure	11
3.2	Observed Logs and Service Behavior	11
3.3	Reverting Changes and Service Restoration	12
4	Task 4: SRE Alert and Issue Analysis	13
4.1	Issue 1: Disk Space Exhaustion	13
4.1.1	Cause	13
4.1.2	Behaviour	13
4.1.3	Resolution	13
4.1.4	Preventive Measures	14
4.2	Issue 2: High CPU Utilization	14
4.2.1	Cause	14
4.2.2	Behaviour	14
4.2.3	Resolution	14
4.2.4	Preventive Measures	15

Chapter 1

Task 1: Telegraf Deployment and Monitoring

1.1 Telegraf Installation

```
1 sudo apt-get update && sudo apt-get install telegraf
```

Listing 1.1: Telegraf installation commands

Explanation

- Lines 1: Helps to install telegraf

1.2 Telegraf Configuration

```
1 # telegraf.conf
2 [global_tags]
3   node_host = "stg-hdpsiddharth101.phonepe.nb6"
4
5 [agent]
6   interval = "60s"
7   round_interval = true
8   metric_batch_size = 2000
9   metric_buffer_limit = 10000
10  collection_jitter = "0s"
11  flush_interval = "60s"
12  flush_jitter = "0s"
13  precision = ""
14  debug = true
15  logtarget = "file"
16  logfile = "/var/log/telegraf/telegraf.log"
17  logfile_rotation_max_size = "25MB"
```

```

18  logfile_rotation_max_archives = 3
19  log_with_timezone = "local"
20  hostname = ""
21  omit_hostname = true
22
23 [[inputs.cpu]]
24   percpu = true
25   totalcpu = true
26   collect_cpu_time = false
27   report_active = true
28
29 [[inputs.mem]]
30
31 [[inputs.elasticsearch]]
32   servers = ["https://10.57.40.168:9200"]
33   username = "elastic"
34   password = "4+fgoJIF16HbF4WxAFlR"
35   tls_ca = "/home/sre/elasticsearch-9.1.3/config/certs/
36     http_ca.crt"
36   cluster_health = true
37   local = false
38   [inputs.elasticsearch.tags]
39     owner = "sid"
40
41 [[inputs.exec]]
42   commands = ["/usr/local/bin/es_log_monitor.py"]
43   timeout = "10s"
44   data_format = "json"
45
46
47 [[outputs.http]]
48   url = "https://metricingestion.nixy.stg-drove.phonepe.nb6/
49     ingestion/v3/telegraf/metrics/bulk"
50   method = "POST"
51   data_format = "json"
52   timeout = "5s"
53   insecure_skip_verify = true
54
55   [outputs.http.headers]
56     Content-Type = "application/json"
      Authorization = "Bearer XXX"

```

Listing 1.2: Telegraf configuration ('telegraf.conf')

Explanation

- **Lines 2–3:** The `[global_tags]` section defines tags that will be added to every metric collected by this Telegraf agent. Here, `node_host` is set to a specific staging hostname.
- **Lines 5–21:** The `[agent]` section configures the Telegraf agent's core be-

havior.

- **Lines 6–12:** Set the timing for metric collection and flushing. Metrics are gathered every 60 seconds (`interval`) and sent to outputs every 60 seconds (`flush_interval`). Jitter is disabled to ensure consistent timing.
- **Lines 8–9:** Configure the agent's buffer, allowing it to hold up to 10,000 metrics in memory and send them in batches of 2000.
- **Line 14:** Enables `debug` mode for verbose logging, which is useful for troubleshooting.
- **Lines 15–19:** Configure file-based logging. Logs are written to `/var/log/telegraf/telegraf`, rotated when they reach 25 MB, and up to 3 old archives are kept.
- **Line 21:** `omit_hostname = true` prevents Telegraf from automatically adding a 'host' tag, as a custom `node_host` tag was already defined in `global_tags`.
- **Lines 23–27:** The `[[inputs.cpu]]` plugin is configured to collect CPU metrics for each core (`percpu`) and as a total (`totalcpu`).
- **Line 29:** The `[[inputs.mem]]` plugin is enabled with its default settings to collect memory usage metrics.
- **Lines 31–40:** The `[[inputs.elasticsearch]]` plugin is configured to collect metrics from an Elasticsearch cluster.
- **Lines 32–35:** Provide the connection details, including the server URL, credentials, and the path to a TLS certificate for a secure connection.
- **Line 36:** `cluster_health = true` enables the collection of cluster status metrics (green, yellow, red).
- **Lines 38–39:** An input-specific tag `owner = "sid"` is added only to metrics coming from this Elasticsearch plugin.
- **Lines 41–44:** The `[[inputs.exec]]` plugin is used to run a custom script.
- **Line 42:** Specifies that the Python script `es_log_monitor.py` will be executed to generate metrics.
- **Line 44:** Telegraf is instructed to parse the script's output as `json`.
- **Lines 47–56:** The `[[outputs.http]]` plugin is configured to send all collected metrics to a custom HTTP endpoint.
- **Line 48:** Defines the destination URL for the metrics.
- **Line 50:** The outgoing metric data is formatted as a `json` payload.
- **Line 52:** `insecure_skip_verify = true` disables SSL certificate validation, which is common in non-production (staging) environments.
- **Lines 54–56:** Set custom HTTP headers required by the endpoint, including the `Content-Type` and an `Authorization` bearer token.

1.3 Starting Telegraf and Verification

```
1 systemctl start telegraf
2 systemctl status telegraf
3
4 # Command to check logs for any errors
5 journalctl -u telegraf -f
6 tail -f /var/log/telegraf/telegraf.conf
7 telegraf --test
```

Listing 1.3: Starting and checking Telegraf service

```
root@stg-hdpsiddharth101:/etc/telegraf[nb6]# systemctl status telegraf
● telegraf.service - Telegraf
   Loaded: loaded (/lib/systemd/system/telegraf.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-09-24 11:28:17 IST; 1h 25min ago
       Docs: https://github.com/influxdata/telegraf
 Main PID: 1886154 (telegraf)
   Tasks: 14 (limit: 38493)
  Memory: 30.9M
    CGroup: /system.slice/telegraf.service
           └─1886154 /usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-directory /etc/telegraf/telegraf.d

Sep 24 11:28:17 stg-hdpsiddharth101 systemd[1]: Starting Telegraf...
Sep 24 11:28:17 stg-hdpsiddharth101 telegraf[1886154]: 2025-09-24T05:58:17Z ! Loading config: /etc/telegraf/telegraf.conf
Sep 24 11:28:17 stg-hdpsiddharth101 telegraf[1886154]: 2025-09-24T05:58:17Z ! Loading config: /etc/telegraf/telegraf.d/cores.conf
Sep 24 11:28:17 stg-hdpsiddharth101 telegraf[1886154]: 2025-09-24T05:58:17Z ! Agent setting "logtarget" is deprecated, please just set "logfile" and remove this setting! The setting will be removed in v1.40.0.
Sep 24 11:28:17 stg-hdpsiddharth101 telegraf[1886154]: 2025-09-24T05:58:17Z W! Agent setting "logtarget" is deprecated, please just set "logfile" and remove this setting! The setting will be removed in v1.40.0.
[root@stg-hdpsiddharth101:/etc/telegraf[nb6]# tail -f /var/log/telegraf.log
2025-09-24T12:49:17+05:30 D [outputs.http] Wrote batch of 35 metrics in 20.08817ms
2025-09-24T12:49:17+05:30 D [outputs.http] Buffer fullness: 0 / 10000 metrics
2025-09-24T12:50:17+05:30 D [outputs.http] Wrote batch of 35 metrics in 20.546402ms
2025-09-24T12:50:17+05:30 D [outputs.http] Buffer fullness: 0 / 10000 metrics
2025-09-24T12:51:17+05:30 D [outputs.http] Wrote batch of 35 metrics in 20.704328ms
2025-09-24T12:52:17+05:30 D [outputs.http] Buffer fullness: 0 / 10000 metrics
2025-09-24T12:52:17+05:30 D [outputs.http] Wrote batch of 35 metrics in 16.438473ms
2025-09-24T12:52:17+05:30 D [outputs.http] Buffer fullness: 0 / 10000 metrics
2025-09-24T12:53:17+05:30 D [outputs.http] Wrote batch of 35 metrics in 16.294495ms
2025-09-24T12:53:17+05:30 D [outputs.http] Buffer fullness: 0 / 10000 metrics
^C
root@stg-hdpsiddharth101:/etc/telegraf[nb6]#
```

Figure 1.1: Screenshot showing Telegraf service status is active.

Chapter 2

Task 2: Grafana Visualization

2.1 Grafana Setup

```
1 sudo apt-get install -y adduser libfontconfig1 musl
2 wget https://dl.grafana.com/grafana/release/12.2.0/grafana_12
   .2.0_17949786146_linux_amd64.deb
3 sudo dpkg -i grafana_12.2.0_17949786146_linux_amd64.deb
4
5 systemctl status grafana-server
```

Listing 2.1: Grafana installation and startup commands

2.2 Adding OpenTSDB as a Data Source

Explanation

Adding the Stage OpenTSDB instance as a data source in Grafana.

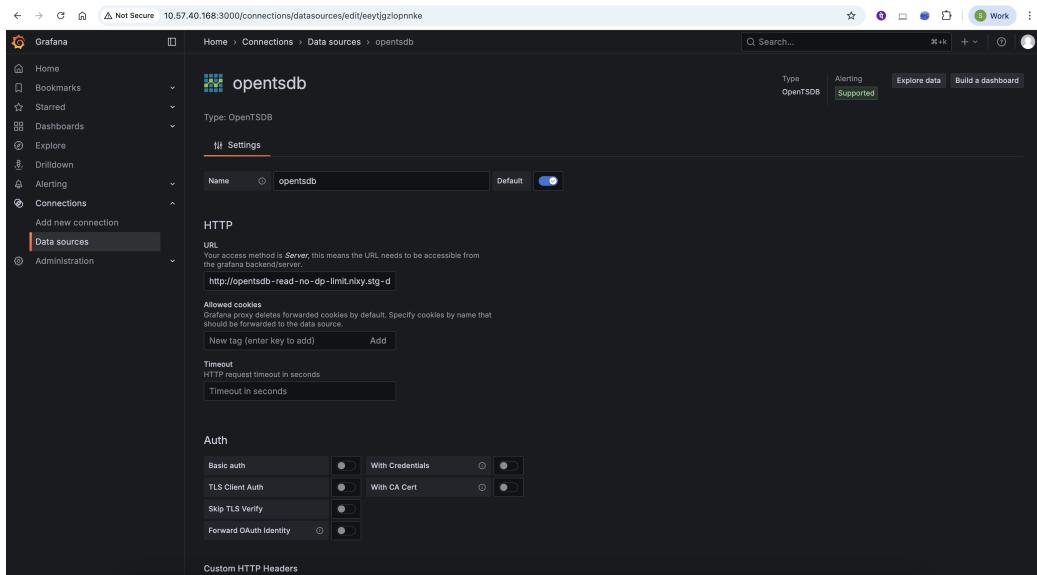


Figure 2.1: Screenshot of the OpenTSDB data source configuration in Grafana.

2.3 Visualizing Metrics in Grafana

Explanation

Overview of the dashboard I created and the metrics I chose to visualize.

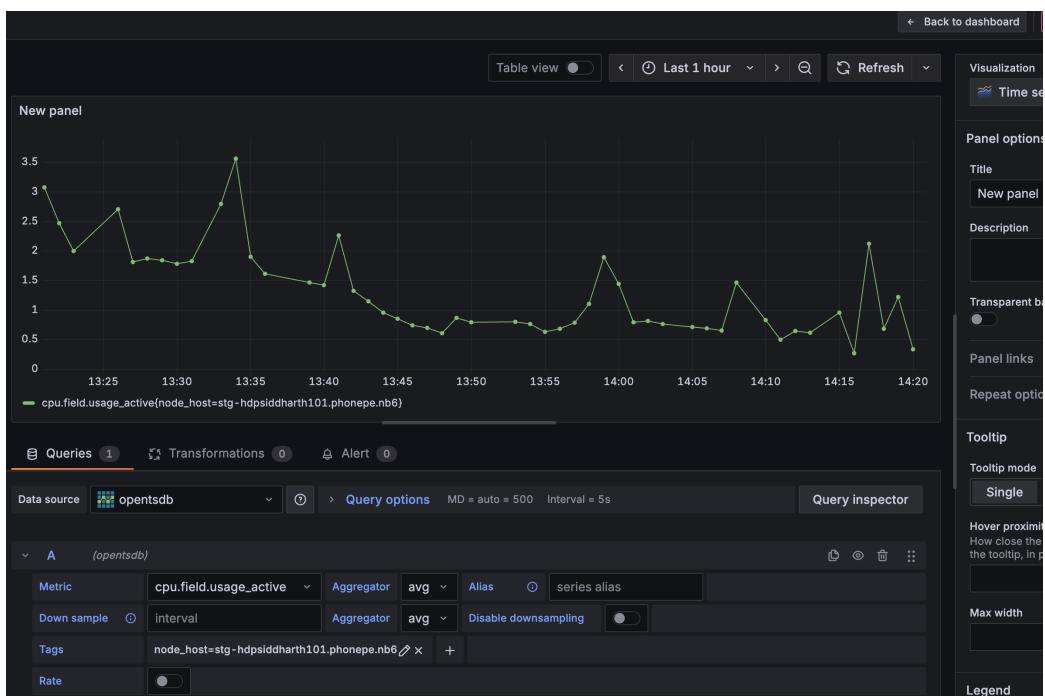


Figure 2.2: Grafana dashboard showing system CPU utilization.

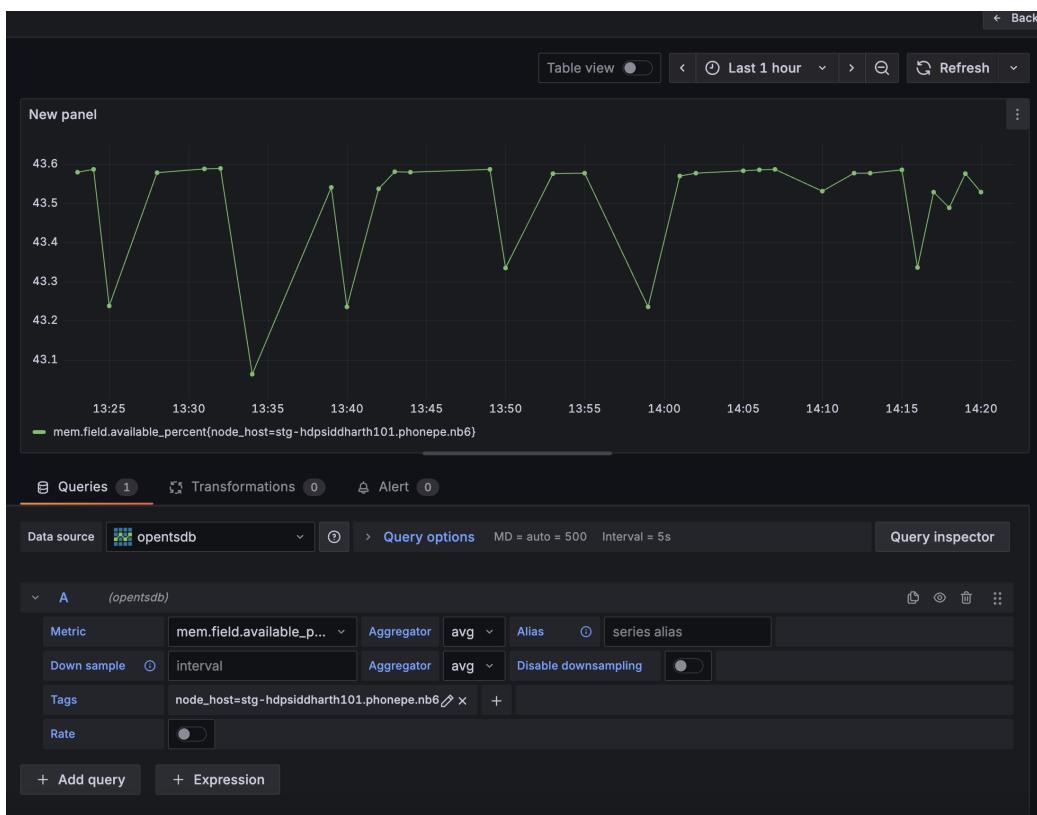


Figure 2.3: Grafana dashboard showing system memory usage.

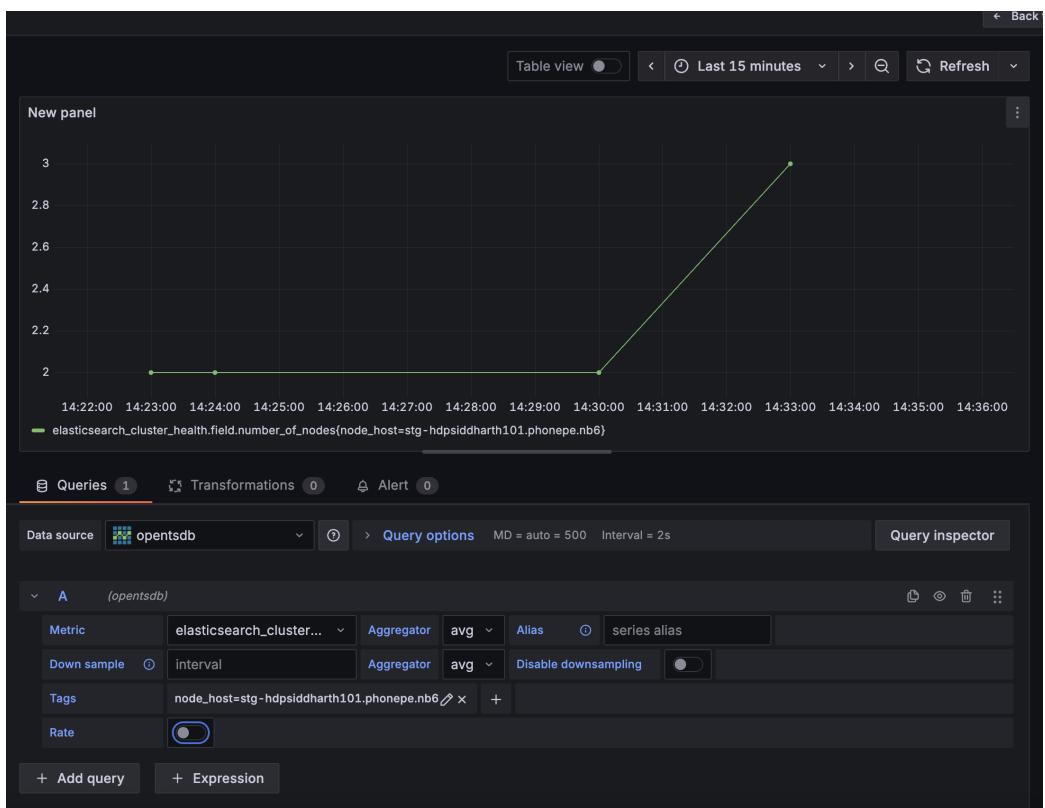


Figure 2.4: Grafana dashboard showing Elasticsearch cluster health status.

Chapter 3

Task 3: Inducing and Observing Service Impact

3.1 Description of Induced Failure

```
1 # Example: Modifying a config and restarting the service
2 "discovery.seed_hosts": ["invalid-host"]
3
4 #restarting this node
5 ./bin/elasticsearch -d -p pid
```

Listing 3.1: Command or configuration change to cause failure

3.2 Observed Logs and Service Behavior

Explanation

Master not found and hence it not get inside the cluster.

```
[2025-09-24T14:41:39.397][INFO][o.e.e.c.r.PermissionCheck] [node3] acquired cache index access
[2025-09-24T14:41:39.397][INFO][o.e.e.c.r.PermissionCheck] [node3] replication component started
[2025-09-24T14:41:39.454][INFO][o.e.e.t.TransportService ] [node3] publish_address {10.57.48.109:9308}, bound_addresses {0.0.0.0:9308}
[2025-09-24T14:41:46.457][INFO][o.e.e.b.BootstrapChecks ] [node3] publishing to a random local IP address, enforcing bootstrap checks
[2025-09-24T14:41:46.457][INFO][o.e.e.b.BootstrapChecks ] [node3] bootstrap check failed, reason: [bootstrap missing master node], will not attempt further cluster bootstrapping
[2025-09-24T14:41:50.874][WARN][o.e.e.c.c.ClusterFormationFailureHelper] [node3] master not discovered or elected yet, an election requires at least 2 nodes with ids from [Pppow2zrSDKRRGZJnD14_Q, f5WtMWhSe-Sra3l0MCnA, 0Mhicy48ACx72mC2tjTQ], h
are only discovered non-augustus [node3] (10.57.48.109) and [node3] (10.57.48.109) from last-known cluster state: node term 10, last accepted version 1687 in term 18; for troubleshooting guidance, see https://www.e
lastic.co/docs/troubleshoot/elasticsearch/discovery-troubleshooting#version9_1
[2025-09-24T14:42:10.877][INFO][o.e.e.c.r.PermissionCheck] [node3] master not discovered or elected yet, an election requires at least 2 nodes with ids from [Pppow2zrSDKRRGZJnD14_Q, f5WtMWhSe-Sra3l0MCnA, 0Mhicy48ACx72mC2tjTQ], h
are only discovered non-augustus [node3] (10.57.48.109) and [node3] (10.57.48.109) from last-known cluster state: node term 10, last accepted version 1687 in term 18; for troubleshooting guidance, see https://www.e
lastic.co/docs/troubleshoot/elasticsearch/discovery-troubleshooting#version9_1
[2025-09-24T14:42:10.877][INFO][o.e.e.c.c.ClusterFormationFailureHelper] [node3] master not discovered or elected yet, an election requires at least 2 nodes with ids from [Pppow2zrSDKRRGZJnD14_Q, f5WtMWhSe-Sra3l0MCnA, 0Mhicy48ACx72mC2tjTQ], h
are only discovered non-augustus [node3] (10.57.48.109) and [node3] (10.57.48.109) from last-known cluster state: node term 10, last accepted version 1687 in term 18; for troubleshooting guidance, see https://www.e
lastic.co/docs/troubleshoot/elasticsearch/discovery-troubleshooting#version9_1
[2025-09-24T14:42:10.877][INFO][o.e.e.c.r.PermissionCheck] [node3] published after [discovery.initial.state.timeout=30s] while waiting for initial discovery state; for troubleshooting guidance see [https://www.elastic.co/docs/troubleshoot/elast
icsearch/discovery-troubleshooting#version9_1]
[2025-09-24T14:42:10.877][INFO][o.e.e.c.r.PermissionCheck] [node3] publish_address {10.57.48.109:9308}, bound_addresses {0.0.0.0:9308}
[2025-09-24T14:42:10.877][INFO][o.e.e.NNode ] [node3] started (node3)(0Mhicy48ACx72mC2tjTQ)(1VV_12fH04_A1vlnTw)(node3)(10.57.48.109)(0.1.3)(8000099-9833080){el.config.version=12.0.0, xpack.installed=true
e, transform.config_version=10.0.0} ... ... =
```

Figure 3.1: Screenshot showing the cluster is not formed.

3.3 Reverting Changes and Service Restoration

Explanation

After reverting changes cluster formed and (BAU - Business As Usual).

Figure 3.2: Screenshot showing the cluster formed again.

Chapter 4

Task 4: SRE Alert and Issue Analysis

4.1 Issue 1: Disk Space Exhaustion

4.1.1 Cause

The primary cause is the uncontrolled growth of data, such as application logs, database files, or temporary files, without proper rotation, cleanup, or capacity planning. In stateful services like Elasticsearch or databases, ever-growing indexes or data can quickly fill up the available disk space.

4.1.2 Behaviour

- The service may fail to start or crash unexpectedly.
- Applications will be unable to write new data, leading to errors (e.g., HTTP 500 errors).
- For systems like Elasticsearch, the cluster health will turn RED, and it will enforce a read-only block on indexes located on the full node.
- System performance may degrade significantly as the OS struggles to manage files on a full disk.

4.1.3 Resolution

- **Immediate:** Identify and delete unnecessary files (e.g., old log files, core dumps, temporary files).
- **For Databases/Elasticsearch:** Delete old or non-critical indexes/data after taking a backup if necessary. Use APIs to clear caches or shrink data files where possible.
- **Mid-term:** Increase the disk size of the affected server or node.

4.1.4 Preventive Measures

- **Monitoring & Alerting:** Set up alerts for disk usage (e.g., alert when usage is greater than 85%, critical alert when greater than 95%).
- **Automation:** Implement automated log rotation and cleanup scripts (e.g., using ‘logrotate’).
- **Capacity Planning:** Regularly review disk usage trends to proactively scale storage before it becomes critical.
- **Data Lifecycle Management:** For services like Elasticsearch, implement Index Lifecycle Management (ILM) policies to automatically move data to cheaper storage or delete it after a certain period.

4.2 Issue 2: High CPU Utilization

4.2.1 Cause

This can be caused by various factors: a sudden spike in user traffic, an inefficient query running in a loop, a software bug causing a process to consume excessive resources, or insufficient hardware resources for the current workload. A misconfiguration, such as setting too many worker threads, can also lead to CPU contention.

4.2.2 Behaviour

- Slow response times for applications and APIs (high latency).
- The server may become unresponsive to SSH or other commands.
- Health checks may fail, causing load balancers to remove the instance from the pool.
- If part of an auto-scaling group, it might trigger the creation of new instances, potentially hiding the root cause.

4.2.3 Resolution

- **Immediate:** Identify the top CPU-consuming processes using tools like ‘top’ or ‘htop’.
- If a known problematic process is identified, restart it.
- If it’s a query-related issue, kill the long-running query at the database/service level.
- If the load is legitimate, scale up the service (add more instances or increase CPU cores).

4.2.4 Preventive Measures

- **Monitoring & Alerting:** Set up alerts for sustained high CPU usage (e.g., greater than 90 % for more than 5 minutes).
- **Performance Testing:** Regularly conduct load testing to understand the service's breaking points and identify performance bottlenecks.
- **Code Optimization:** Implement efficient algorithms and database queries. Use caching where appropriate to reduce computational load.
- **Auto-scaling:** Configure proper auto-scaling policies based on CPU utilization to handle traffic spikes gracefully.