**Solution for Lab#05:**

# Observations:

|  |  |  |  |
| --- | --- | --- | --- |
| Metric Name | Value | Time Stamp | Description |
| node\_cpu\_seconds\_total | 35421.25 | 2025-04-06T10:20:00Z | Total CPU time in secs |
| node\_memory\_Active\_bytes | 8723942400 | 2025-04-06T10:20:10Z | Active memory in bytes |
| up | 1 | 2025-04-06T10:20:15Z | Indicates target is UP |

# Result:

Prometheus was successfully installed and configured. Using Node Exporter, we monitored system performance metrics including CPU usage and memory status. The data was visualized in the Prometheus web interface using PromQL queries. Optional Grafana integration was available for enhanced visualization.

# Viva Questions – Answered:

* Q: What is Prometheus used for?

A: Prometheus is used for monitoring and alerting. It collects, stores, and visualizes time-series metrics data from targets like servers or containers.

* Q: What is a time-series database?

A: A time-series database stores data points indexed by time. It is ideal for tracking changes over time, such as CPU load or memory usage.

* Q: Explain PromQL with an example.

A: PromQL is Prometheus’s query language. Example: `node\_cpu\_seconds\_total` – returns total CPU usage in seconds.

* Q: What is the role of exporters in Prometheus?

A: Exporters expose system/application metrics in a format Prometheus can scrape. E.g., Node Exporter exposes system-level metrics.

* Q: How do you configure a new target in Prometheus?

A: In the `prometheus.yml` file under `scrape\_configs`, you define a `job\_name` and list of `targets` with their host and port.

# References:

https://prometheus.io

https://grafana.com

Prometheus Official Documentation