

SET-222

Software Operations & Maintenance

Experiment # 05

**Experiment Title**

**Monitoring Software Performance with Prometheus**

**Assessment of CLO(s): 03**

**Performed on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | **Group** |  |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Ms. Shagufta Aftab** | | |
| **Date** |  | **Signature** |  |

## Copyright © Department of Engineering & Technology – UIT University Karachi

**Objective:**

 To understand the fundamentals of Prometheus.

 To install and configure Prometheus for monitoring.

 To monitor software performance metrics using Prometheus.

 To visualize data using Prometheus's web UI and/or Grafana (optional).

**Theory:**  
**Prometheus** is an open-source monitoring and alerting toolkit designed for reliability and scalability. It collects and stores metrics as time-series data, recording information with a timestamp and optional key-value pairs (labels).

**Key Concepts:**

* **Time-Series Data**
* **Exporters** (e.g., Node Exporter)
* **PromQL** (Prometheus Query Language)
* **Targets & Jobs**
* **Alertmanager** (for alerting)
* **Visualization** through Prometheus UI or Grafana

**4. Procedure:**

**Step 1: Install Prometheus**

You can install Prometheus directly or via Docker.

**Using Docker:**

docker run -p 9090:9090 prom/prometheus

**Or manually:**

1. Download from https://prometheus.io/download/
2. Extract and navigate to the Prometheus folder.
3. Run:

./prometheus --config.file=prometheus.yml

**Step 2: Install Node Exporter (for system metrics)**

**Using Docker:**

docker run -d -p 9100:9100 prom/node-exporter

**Manually:**

1. Download from https://prometheus.io/download/#node\_exporter
2. Run:

./node\_exporter

**Step 3: Configure Prometheus to Scrape Node Exporter**

Edit prometheus.yml and add:

scrape\_configs:

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['localhost:9100']

Restart Prometheus after saving the config.

**Step 4: Access Prometheus Dashboard**

Open your browser and go to:  
 http://localhost:9090

Try some **PromQL** queries like:

* up
* node\_cpu\_seconds\_total
* node\_memory\_MemAvailable\_bytes

**Step 5 (Optional): Integrate with Grafana**

1. Install Grafana.
2. Add Prometheus as a data source.
3. Import a system monitoring dashboard.
4. Visualize real-time performance metrics.

**5. Observations:**

| **Metric Name** | **Value** | **Time Stamp** | **Description** |
| --- | --- | --- | --- |
| node\_cpu\_seconds\_total | ... | ... | CPU usage |
| node\_memory\_Active\_bytes | ... | ... | Active memory |
| up | 1 | ... | Target status |

**6. Result:**

Prometheus was successfully installed and configured to monitor system performance. System metrics were retrieved and visualized using Prometheus's built-in web UI.

**7. Viva Questions:**

1. What is Prometheus used for?
2. What is a time-series database?
3. Explain PromQL with an example.
4. What is the role of exporters in Prometheus?
5. How do you configure a new target in Prometheus?

**8. References:**

* <https://prometheus.io>
* <https://grafana.com>
* Prometheus Documentation