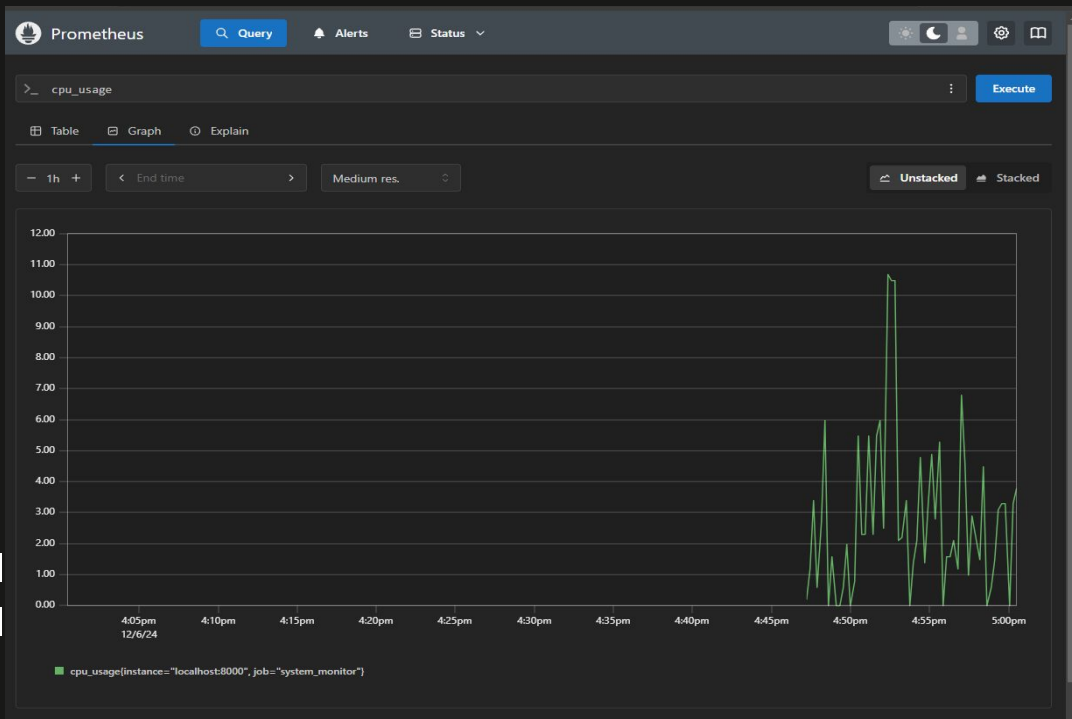


System Monitoring Project with Python, Prometheus, and Grafana

Real-Time System Monitoring and Visualization



Jyotshna Trivedi
jyotshna88088@gmail.com

GitHub:
github.com/dev-jyotshna

Why Monitoring Matters

Systems need constant monitoring for optimal performance and reliability

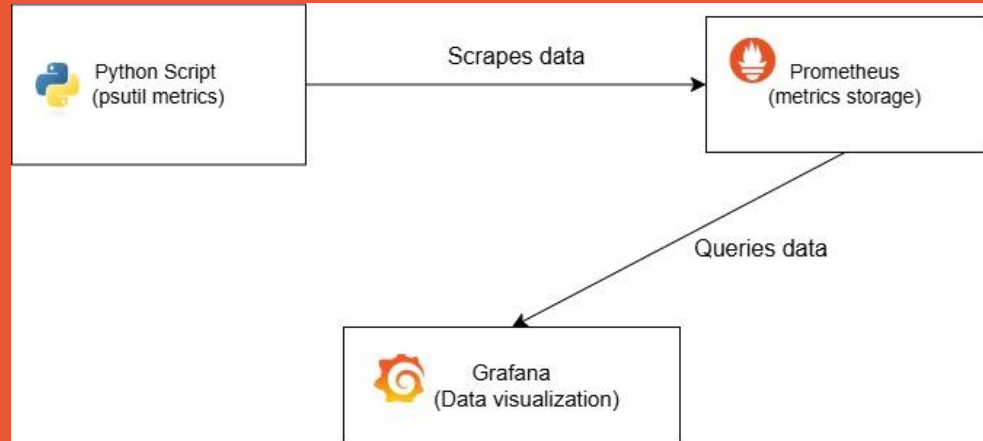
- Commercial monitoring tools can be expensive and inflexible
- A free, open-source alternative is desirable for smaller organizations and personal projects

Visual : Grafana dashboard



Project Goals

- Collect system metrics like CPU, memory, and disk usage in real time
- Store and query metrics efficiently using Prometheus
- Visualize data with Grafana dashboards
- Keep the project cost-free and lightweight



System Architecture

Python Script

Python Script:

Collects system metrics using the psutil library and exposes them via an HTTP endpoint.

This is the initial step in the system architecture where data is gathered.

Prometheus

Database:

Scrapes the metrics from the Python script and stores them in its time-series database.

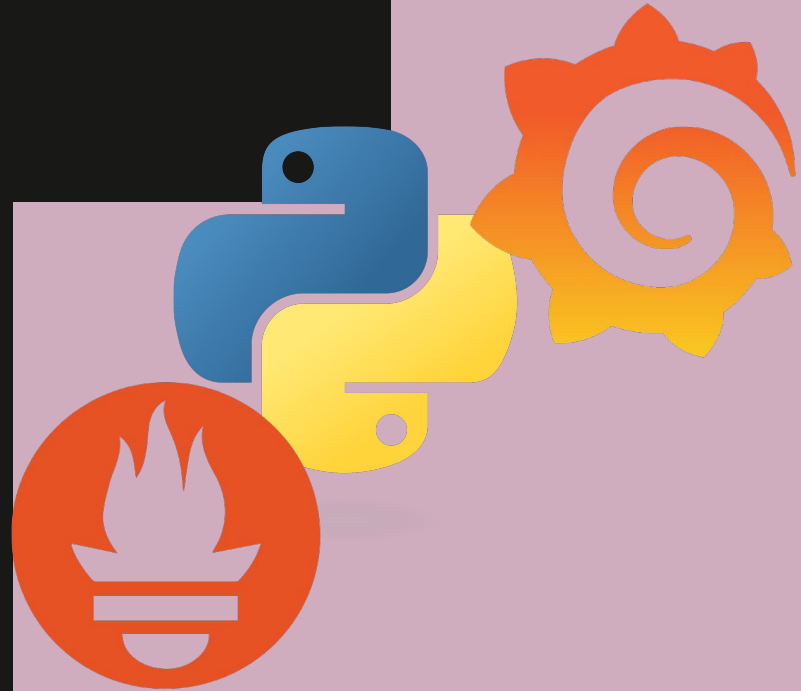
Grafana

Visualization:

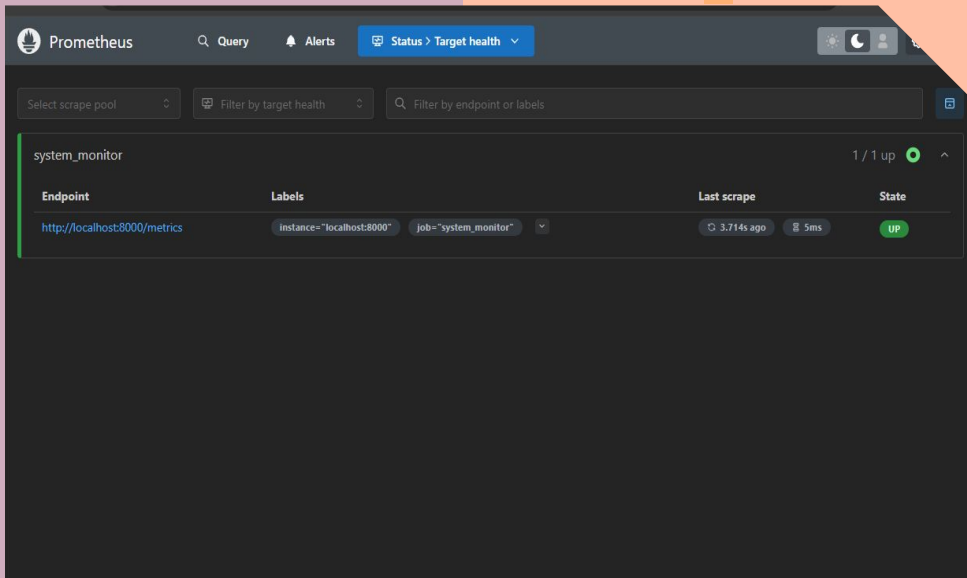
Queries Prometheus and visualizes the metrics on interactive dashboards.

Technology Stack

- **Python & psutil:** Collect metrics like CPU, memory, and disk usage
- **Prometheus:** Store and query metrics efficiently
- **Grafana:** Build customizable dashboards



How It Works



The Python script collects system metrics using psutil

- The script exposes these metrics to Prometheus via an HTTP endpoint
- Prometheus scrapes and stores the metrics
- Grafana queries Prometheus and displays the data visually

```
def monitor_system():
    """Monitor and expose metrics while handling alerts."""
    while True:
        # Get CPU, memory usage, disk usage
        cpu_usage = psutil.cpu_percent(interval=1)
        memory_usage = psutil.virtual_memory().percent
        disk_usage = psutil.disk_usage(DISK_PATH).percent

        # Update Prometheus metrics
        cpu_usage_gauge.set(cpu_usage)
        memory_usage_gauge.set(memory_usage)
        disk_usage_gauge.set(disk_usage)

        if cpu_usage > CPU_THRESHOLD:
            log_event(f"High CPU Usage: {cpu_usage}%")
            send_email("High CPU Usage Alert", f"CPU usage is at {cpu_usage}%")

        if memory_usage > MEMORY_THRESHOLD:
            log_event(f"High Memory Usage: {memory_usage}%")
            send_email("High Memory Usage Alert", f"Memory usage is at {memory_usage}%")

        if disk_usage > DISK_THRESHOLD:
            message = f"High Disk Usage on {DISK_PATH}: {disk_usage}%"
            log_event(message)
            send_email("High Disk Usage Alert", message)

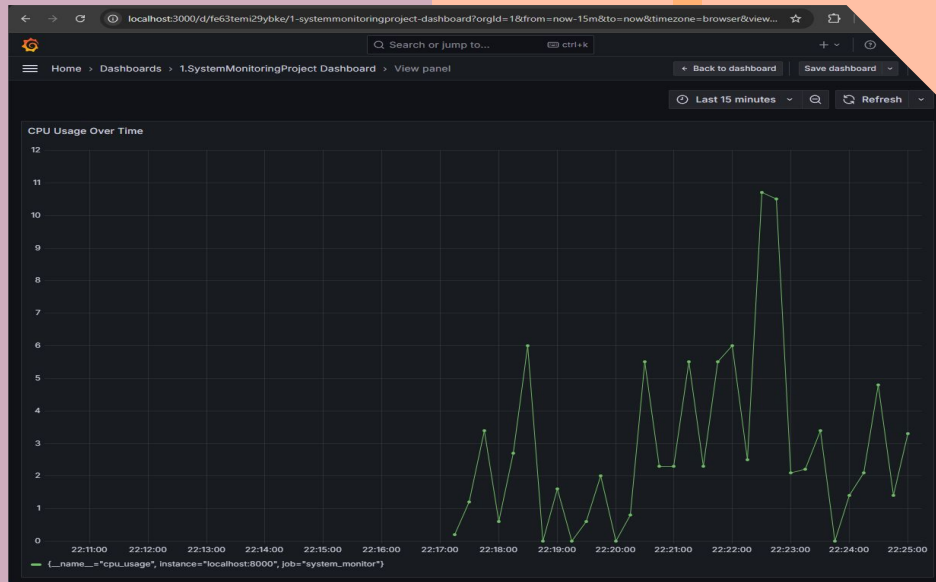
        print(f"CPU Usage: {cpu_usage}%, Memory Usage: {memory_usage}%, Disk Usage: {disk_usage}%")

        time.sleep(5)

if __name__ == "__main__":
    start_http_server(8000)
    print("Prometheus metrics available at http://localhost:8000/metrics")
    monitor_system()
```

Project in Action

- Run the Python script to expose metrics
- Start Prometheus and verify the target
- Open Grafana and view the live dashboard



Key Features

- Real-time system monitoring for CPU, memory, and disk usage
- Easy-to-extend architecture for adding new metrics
- User-friendly Grafana dashboards
- Completely free and open-source

Overcoming Challenges

- **Configuring Prometheus** to scrape data at proper intervals
 - → Resolved by editing the prometheus.yml file
- **Designing clear Grafana dashboards**
 - → Iteratively tested with sample data
- **Managing sensitive credentials**
 - → Used .env files and .gitignore

```
scrape_configs:  
  # The job name is added as a label `job=<job_name>`  
  - job_name: "system_monitor"  
  
    # metrics_path defaults to '/metrics'  
    # scheme defaults to 'http'.  
  
  static_configs:  
    - targets: ["localhost:8000"]
```

Enhancements and Scalability

Future Scope

- Add more metrics (e.g., network bandwidth, GPU usage)
- Deploy the system on the cloud (AWS/ GCP/ Azure)
- Implement alerting and notifications for multiple and critical thresholds
- Optimize the Python script for distributed environments

THANK YOU

Additional Resources

- **GitHub Repository:** github.com/dev-jyotshna/system-monitoring-project
- **Prometheus Documentation:** [Prometheus](https://prometheus.io/docs/)
- **Grafana Documentation:** [Grafana](https://grafana.com/docs/)
- **psutil Documentation:** [psutil](https://psutil.readthedocs.io/en/latest/)

GitHub repo QR Code

