PCET's
Pimpri
Chinchwad
University, Pune

Learn | Grow | Achieve

Checked By	Dr. Avinash Golande
Sign	
Date	
Grade	

EXPERIMENT NO . 06

- **Title:-** Develop a resource monitoring script in Python using **psutil** to monitor CPU, memory, and disk usage, simulating a cloud monitoring mechanism.

- **Objective :-**

  - To understand the concept of **cloud monitoring**.
  - To implement a Python script for monitoring CPU, memory, and disk usage.
  - To simulate **cloud monitoring alarms** (alerts when usage crosses thresholds).

- **Resources used :-** PC / Laptop with Linux OS (Ubuntu/Debian recommended) , psutil library (pip install psutil) , time, datetime modules

- **Theory :-**

Cloud computing platforms rely on continuous monitoring of system resources to ensure applications run smoothly and efficiently. Since cloud servers and virtual machines handle dynamic workloads, resource utilization such as CPU, memory, and disk must be tracked in real time. If these resources become over-utilized, applications may experience degraded performance, high latency, or even system crashes.

To avoid such issues, organizations use cloud monitoring tools like AWS CloudWatch, Azure Monitor, and Datadog. These tools continuously collect performance metrics, generate logs, provide dashboards for visualization, and trigger alerts when thresholds are exceeded. Alerts notify system administrators to take corrective action, such as scaling resources, optimizing processes, or balancing workloads.

In this experiment, we simulate the concept of cloud monitoring at a local system level using the Python psutil library. The psutil (Process and System Utilities) module provides an interface for retrieving information on running processes and system utilization, including CPU, memory, and disk usage. By combining it with Python’s built-in time and datetime modules, we can log system states with timestamps and check whether usage crosses predefined thresholds.

The monitoring focuses on three key parameters:

- CPU Usage (%): Percentage of processor utilization.
- Memory Usage (%): RAM consumption level.
- Disk Usage (%): Storage utilization percentage.
- In this practical, thresholds are defined as follows:
- CPU usage > 80%
- Memory usage > 80%
- Disk usage > 85%

If any of these limits are exceeded, the script simulates cloud monitoring alarms by generating alerts. This mimics the behavior of enterprise cloud monitoring solutions, where administrators are informed in real time to maintain performance and reliability.

Thus, the implementation of this resource monitoring script not only provides hands-on experience with Python and the psutil library but also demonstrates the core principles of cloud monitoring mechanisms in a simplified local environment.

Concept of Cloud Monitoring

- Cloud servers and virtual machines handle dynamic workloads.

If CPU, memory, or disk are over-utilized, apps may slow down or crash.

Cloud monitoring tools (AWS CloudWatch, Azure Monitor, Datadog, etc.) continuously check resource usage.

- They provide:

Logs – system state with timestamps.

Alerts – warnings when usage crosses thresholds.

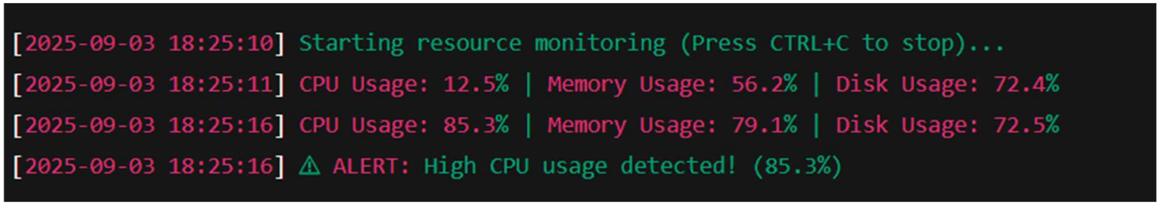
Dashboards – visualization of performance.

In this practical, we simulate local monitoring using **Python psutil library**.

- ## Code

```
import psutil

import time
import datetime

# Thresholds for alerts (simulating cloud monitoring alarms)
CPU_THRESHOLD = 80.0     # in %
MEMORY_THRESHOLD = 80.0  # in %
DISK_THRESHOLD = 85.0    # in %

def log_message(message):
    """Log monitoring messages with timestamps."""
    print(f"[{datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')}] {message}")

def monitor_resources(interval=5):
    """Monitor CPU, memory, and disk usage at regular intervals."""
    log_message("Starting resource monitoring (Press CTRL+C to stop)...")

    try:
        while True:
            # CPU Usage
            cpu_usage = psutil.cpu_percent(interval=1)

            # Memory Usage
            memory = psutil.virtual_memory()
            memory_usage = memory.percent

            # Disk Usage
            disk = psutil.disk_usage('/')
            disk_usage = disk.percent

            # Print resource stats
            log_message(f"CPU Usage: {cpu_usage}% | Memory Usage: {memory_usage}% | Disk Usage: {disk_usage}%")
            # Check alerts
            if cpu_usage > CPU_THRESHOLD:
                log_message(f"⚠ ALERT: High CPU usage detected! ({cpu_usage}%)")
            if memory_usage > MEMORY_THRESHOLD:
                log_message(f"⚠ ALERT: High Memory usage detected! ({memory_usage}%)")
            if disk_usage > DISK_THRESHOLD:
                log_message(f"⚠ ALERT: High Disk usage detected! ({disk_usage}%)")

            time.sleep(interval)  # wait before next check

    except KeyboardInterrupt:
        log_message("Monitoring stopped by user.")

# Run the monitoring script
if __name__ == "__main__":
    monitor_resources(interval=5)
```

- **Output**



```
[2025-09-03 18:25:10] Starting resource monitoring (Press CTRL+C to stop)...
[2025-09-03 18:25:11] CPU Usage: 12.5% | Memory Usage: 56.2% | Disk Usage: 72.4%
[2025-09-03 18:25:16] CPU Usage: 85.3% | Memory Usage: 79.1% | Disk Usage: 72.5%
[2025-09-03 18:25:16] ⚠ ALERT: High CPU usage detected! (85.3%)
```

- **Conclusion:**
    - Successfully developed a Python script using **psutil** for monitoring system resources.
    - Simulated **cloud monitoring alarms** by generating alerts when thresholds were crossed.
    - Learned how cloud monitoring works to ensure performance and reliability in distributed systems.