|  |  |  |
| --- | --- | --- |
| SCHOOL OF INFORMATION AND TECHNOLOGY | | |
| NAME: BUSONGAN, DEMIRAYE-ANNE | DATE PERFORMED: 16 OCT 24 |  |
| Section: IDC1 | DATE SUBMITTED: 16 OCT 24 |

# SYSADM1 – Web Server Monitoring

1. How do you monitor web server statistics?

According to NETDATA, monitoring web server statistics is crucial for optimal performance and user satisfaction. Here’s a streamlined approach:

1. **Choose Monitoring Tools**: Select tools like Netdata for real-time insights, Prometheus and Grafana for advanced metrics, or Nagios/Zabbix for alerts and health monitoring.
2. **Collect Host Metrics**: Monitor key host metrics, including uptime, CPU and memory usage, and disk I/O to prevent overload and bottlenecks.
3. **Monitor Web Server Metrics**: Focus on request and response rates, error codes, response times, response sizes, and active connections to assess server health.
4. **Utilize Built-in Modules**: Use Apache's mod\_status or NGINX's ngx\_http\_stub\_status\_module for direct performance metrics.
5. **Set Up Log Monitoring**: Implement log analysis tools to track traffic patterns and errors, extracting useful metrics like request paths.
6. **Create Dashboards and Alerts**: Develop customizable dashboards for key metrics and set alert thresholds to notify sysadmins of issues.
7. **Analyze and Optimize**: Regularly review data to identify trends, adjust resources, and optimize configurations based on usage patterns.
8. **Conduct Regular Reviews**: Continuously evaluate your monitoring strategy to ensure it meets the evolving needs of your infrastructure.

By following these steps, you can effectively monitor your web servers for reliable and efficient performance.  
  
Retrieved October 16, 2024 from https://www.netdata.cloud/blog/web-servers-and-their-performance/

1. What are the key metrics that you need to monitor in a web server?

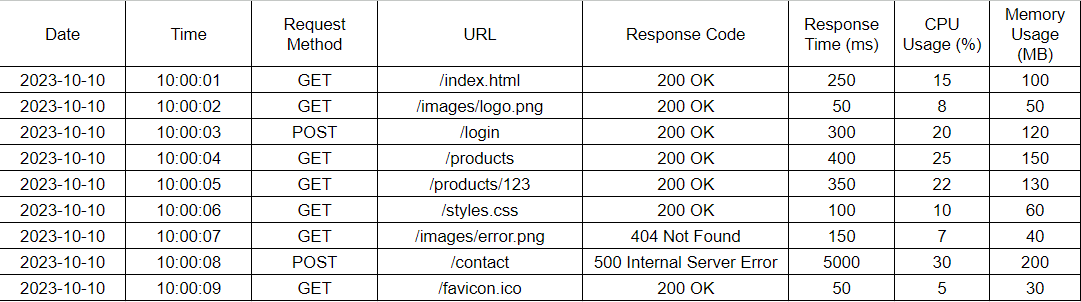
According to NETDATA, key metrics to monitor in a web server include:

1. **Request Rate**: The number of incoming requests over a specific time period, indicating traffic levels.
2. **Response Rate**: The number of requests successfully processed by the server, helping to assess server performance.
3. **Error Codes**: HTTP status codes (e.g., 404, 500) that indicate issues in handling requests and can highlight underlying problems.
4. **Response Time**: The time taken to respond to requests, with higher times signaling potential performance issues.
5. **Response Size**: The amount of data returned in responses, which can impact load times and bandwidth usage.
6. **Active Connections**: The number of current connections being processed, helping to prevent server overload.
7. **CPU Usage**: The percentage of CPU resources being utilized, important for assessing server load.
8. **Memory Usage**: The amount of RAM in use, which can indicate potential exhaustion or memory leaks.
9. **Disk I/O**: The read/write operations on disk, helping to identify bottlenecks in storage.
10. **Uptime**: The total time the server has been operational, critical for measuring reliability.

Monitoring these metrics helps ensure optimal server performance and user satisfaction.

Retrieved October 16, 2024 from https://www.netdata.cloud/blog/web-servers-and-their-performance/

1. Analyze the provided web server statistics to determine what is being asked for below.



1. Average response time: **738.8888889(ms)**
2. Request per second: **1 request per second**
3. Memory usage: **97.77777778%**
4. Error rate: **22.22%**
5. Common error types: **404 Not Found**

**500 Internal Server Error**

1. What are the possible issues in the web server statistics above?  
   To delve into the web server statistics, the high average response time suggests potential performance bottlenecks. The memory usage nearing 100% indicates resource strain, which could lead to crashes or slowdowns. The error rate of around 22% is concerning, especially with the presence of common errors like 404 and 500, pointing to possible issues in routing or server configuration. The 404 error means that the requested resource was not found on the server, indicating a broken link or missing page. The 500 error is a generic server error, suggesting an internal problem that prevented the server from fulfilling the request. Both errors can impact user experience significantly.

**Grading Rubric**

|  |  |  |
| --- | --- | --- |
| Criteria | Points | Description |
| Monitoring methods | 10 | Demonstrates understanding of various methods for monitoring web server statistics (e.g., using tools, server logs, performance counters). |
| Key Metrics | 10 | Identifies the correct key metrics to monitor (e.g., response time, traffic, error rates, resource usage). |
| Data Analysis | 30 | Accurately calculates average response time, requests per second, memory usage, error rate, and identifies common error types. |
| Issues Identification | 10 | Accurately identifies potential issues based on the analyzed statistics (e.g., high error rates, resource constraints). |
| Total | /60 |  |