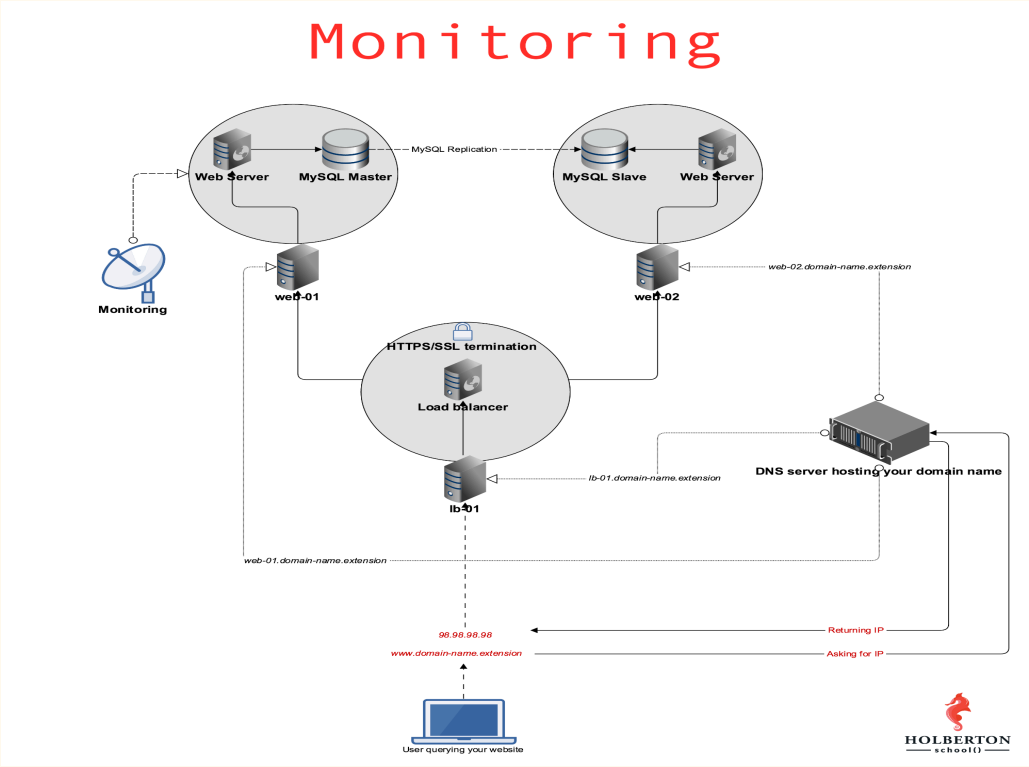
DevOps CD1300: Observability Basics
Holberton Coding School - Tulsa

Observability Basics: Full Stack Monitoring

By Courtney M. Brown



INTRODUCTION

Monitoring technical infrastructure is significant to the security, growth and success of any company or service who hosts their services via web servers. Through observability and monitoring, entities are able to observe the technical stack of each component conducive to the company's technology (usually on a cloud-based service) in real-time. Using the three pillars of observability- metrics, logs, and traces- in combination with application and server monitoring, companies are able to holistically improve their performance and security.

Full Stack Monitoring

Full stack monitoring is the practice of monitoring each technical stack of a cloud-based services' application and servers. Monitoring could be broken down into two categories:

1. **Application Monitoring:** analyzes data from the application's performance, health and behavior.
2. **Server Monitoring:** analyzes data from virtual or physical servers to ensure security and proper load balancing.

Application and server monitoring combined is vital to the overall success of a client's interaction with a company's technology. Full stack monitoring simultaneously analyzes long-term trends in growth over time, tests the runtime of applications and transactions, alerts the developers when a service is or will be broken, builds dashboards for the services provided, and assists developers in debugging. Together, these components of monitoring assist developers and businesses improve their technologies' response time and services, continually.

Application Monitoring

Application monitoring's main objective is to improve and maintain the success of a business' performance and visibility. **Application Performance Monitoring** (APM) is a concept of application monitoring that refers to a collection of software tools that uses telemetry data to ensure the success of a business' technical performance. The APM concept splits this performance monitoring into two sets of metrics:

1. **Load balancing and response times**, which monitors the volume of traffic processed by the application, and the time it takes to respond to client requests. These performance metrics define the user experience.
2. **Computational resources** that test whether the site can support the load or traffic it receives. It can also track performance bottlenecks (CPU and memory usages, I/O's, queries, etc.), which occurs when the rate of accessibility to system data is not compatible with the system's requirements.

There are numerous APM tools to help monitor an application that differ in services and features available to the developers. Some of these tools include Dynatrace, Datadog, Instana, and LogRocket- just to name a few. The common goal within APM software tools is to collect telemetric data and track the performance and runtime of a software or web application in order

to assist with solving any technical issues that arise, or may occur in the near future. APM tools also create graphic metrics for administrators to visualize data and better understand any performance issues.

Server Monitoring

Server monitoring is an absolute necessity for organizations whose services depend on web, network, or application servers that are deployed in the cloud. The key objective of server monitoring can change depending on the **key performance indicator** (KPIs), which is based on the type of server used in an instance. However, the overall goal of server monitoring is to protect the server at all costs in order to ensure security and service availability. Server monitoring can also improve application monitoring by collecting the necessary metrics for good server health and analyzing event logs in areas of the server that promote application performance. The workflow for server monitoring varies depending on the cloud-based service provider (and the tools compatible with the respective platform), the type of server, and the number of servers used to host an application. According to sumologic.com, the general process of server monitoring can observed in five steps:

1. **Identification of what data to track on the server and labeling them as KPIs.** For instance, application servers may want to monitor responsiveness and availability, while web servers monitor speed and loads.
2. **Measuring the performance of each server by implementing a metric system for each KPI**. Organizations may set these metrics as the "baseline" in which the server's data is tested against to ensure overall success of each request.
3. **Creating and configuring a system to pull data from the servers and loading that data into event logs (also referred to as 'log files').** These event logs collect the necessary metrics from servers and will compare them to the baseline. Some metrics collected may include CPU and memory usage, disk performance, network connectivity, etc.
4. **Building a system to alert administrators when there is a problem (or potential problem) with any KPIs.**
5. **Create policies for when issues or threats to the server may arise.** This process will vary on the type of business and services provided, as well as the clients affected by the situation at hand.

Tools used for server monitoring include SolarWinds, Paessler PRTG, Datadog, Atera, and Sematext Infrastructure Monitoring. The common goal of these tools is to measure and display the data of the servers health against the organization's metric baseline, and improve the security of data held on these servers.

Access Logs for Web Servers

An access log is a log file that records any and all requests made to the API. These logs usually include notes about any requests to the API/server including the viewed HTML files and their embedded graphic images, request type (GET, POST, etc.), the kind of device and IP address used for request, users/bots who received a response from the request, server response time, and HTTP status codes which displays the results of each request. Although the type of access log may vary depending on the metrics collected by the organization, the general purpose is to give administrators insight on the improvements to be made to the server and/or application.

Moreover, the data collected in the access logs can assist developers with improving the reliability of the site itself, and in analyzing metrics (overall site visitation, what pages were visited the most, time spent on certain pages, etc.) for marketing purposes. The access logs can also be used to trace the user experience on an application or site. Brand developers and marketers can use this information to improve the user experience, and software developers can use the same information to track trends on visitation (which, in turn, can be used to preload certain pages) and technical trends within the application.

Further, troubleshooting can be conducted easily through analyzing the access logs. Developers can trace trends of failure between software tools and the success of each request. This assists in identifying the issues that need attention, long-term and short-term. Access logs can also expose data breaches and prevent attacks on the server, and more importantly, client data.

Servers like Apache and Nginx have software installed that automatically collects and stores access logs, although the location of such files may vary depending on the server and operating systems used. The key to understanding access logs is to understand the services taking place on a server or application/site, and how they are interpreted in these files.

Error Logs

Similar to access logs, error logs collect the standard errors about failures or encountered issues within the server or application. The different issues that are collected in the error logs range from warnings, errors, critical threats, alerts, and emergencies- all self explanatory in their level of severity. The default setting of error logs work globally, however, one could override this by placing the 'error_log' directive (when using Nginx) in the main (or top-level) configuration context. In addition to the access logs, error logs can also be helpful in detecting troubleshooting problems within a software platform.

Summary

Software tools have been developed so much that they are able to detect problems within a technical infrastructure using a set baseline of metrics, tracking of data that cause issues (or may cause issues in the future), and logging all events that happen on a server. With full stack monitoring, developers and other professionals can quantify performance and improve services based on this data over time.

Resources

https://sre.google/sre-book/monitoring-distributed-systems/

https://docs.nginx.com/nginx/admin-guide/monitoring/logging/

https://en.wikipedia.org/wiki/Application_performance_management

https://www.sumologic.com/glossary/server-monitoring/

https://www.g2.com/categories/application-performance-monitoring-apm

https://docs.oracle.com/cd/E93177_01/docs.83/Alarms%20and%20KPIs/GUID-F68A7508-FDB3-4578-8BFE-078A9F5DB73B.htm#:~:text=Key%20Performance%20Indicators%20(KPIs)%20allow,resides%20on%20all%20OAM%20servers.

https://10web.io/glossary/access-logs/

https://aws.amazon.com/what-is/application-performance-monitoring/

https://www.sumologic.com/glossary/telemetry/#:~:text=analytics%2C%20and%20strategy-,What%20is%20Telemetry%3F,and%20control%20the%20remote%20system.

https://www.mezmo.com/learn-observability/what-is-full-stack-monitoring

https://sematext.com/glossary/full-stack-observability/#:~:text=Full%20stack%20observability%20platforms%20enable,a%20problem%20for%20your%20customers.