Summary

IBM Application Performance Management (APM) monitors, analyzes and manages applications and IT infrastructure. It monitors and analyzes application performance, issues, root causes, and outages to improve user experience and stability. It allows DevOps teams to monitor performance, availability and response times of applications allowing them to identify, isolate and diagnose any existing issues or potential problems.

GitLab has a powerful monitoring capability with Prometheus, a time-series monitoring service, providing a flexible platform for monitoring GitLab and other software products. GitLab provides out-of-the-box monitoring with Prometheus, providing easy access to high-quality time-series monitoring of GitLab services. GitLab has built-in monitoring that can automatically monitor your deployed applications, with extensive support for native cloud, container and microservices technology. Additionally, GitLab uses Jaeger, an open source end-to-end distributed tracing system used for monitoring and troubleshooting microservices-based distributed systems.

Resources

- IBM APM

Feature Comparison

**FEATURES**

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>GitLab</th>
<th>IBM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cloud Native Monitoring</strong></td>
<td><img src="#" alt="Green" /></td>
<td><img src="#" alt="Green" /></td>
</tr>
</tbody>
</table>

The monitoring of cloud native applications including microservices that are built to run in the cloud so as not to bottleneck and issues can be addressed via insights into collected metrics.
Server Monitoring

Reviewing and analyzing a server for availability, operations, performance, security and other operations-related processes. Monitor servers system resources like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Process, etc. GitLab uses the Node Exporter (via Prometheus) to expose an extensive set of machine-level metrics on Linux and other Unix systems such as CPU usage, memory, disk utilization, filesystem fullness, and network bandwidth.

Tracing

Tracing provides insight into the performance and health of a deployed application, tracking each function or microservice which handles a given request. This makes it easy to understand the end-to-end flow of a request, regardless of whether you are using a monolithic or distributed system.

Learn more about Tracing