A Step Ahead in Operational Resilience - Integrating Observability with DevOps

Amit Sengupta (Independent Research)

Email – amits2913@gmail.com
Independent Researcher

Abstract

The finance market closely depends on translation and high-quality software solutions when performing crucial transactions and processing important information and customer services. Thus, systems' reliability and good performance become crucial when these systems become complicated. This paper aims to focus on the implementation of the observability concept with the DevOps approach in financial services technologies, where its strengths, weaknesses, opportunities, and threats are also discussed with regard to the future. The concept of observability is intertwined with DevOps since, with its help, it is possible to gain deep insights into the system's inner state and further enhance status monitoring, detect problems in less time, and optimize performance constantly. When organized and analyzed properly, observability data can, therefore, play a critical role in increasing software quality in financial institutions, aligning with regulatory standards, and decreasing development and operations teams' silos. However, the implementation of observability within an organization using DevOps best practices in the financial services industry has some challenges, which includes the issue of security, especially when it comes to data, the Challenge of data overload, the challenging task of encouraging the right organizational culture for continuous and consistent observability. The article presents a guide that discusses how to incorporate observability with DevOps: the step-by-step process of defining observability needs, choosing the most suitable tools, integrating with other tools in the existing DevOps frameworks, laboratory of alarms, and constant enhancement. Furthermore, it considers examples of how some financial organizations have applied observability to reduce risks, improve efficacy, and enrich customers' interactions. By adopting observability and aligning it with DevOps, financial institutions can develop and sustain sound, reliable and high-quality infrastructure and maintain business continuity.

Keywords: - *IT Service Management, Cost Optimization, Operational Resilience, Integrated Monitoring, Service Operations, Operational Automation, Observability, Monitoring, DevOps, Operational Resilience*

Introduction: -

Most of the financial service functions are dependent on software to enable transaction processing, data management or the delivery of services to consumers. While these systems continue to become intricate, it becomes crucial to establish their sound development and system functionality. Due to such consequences, disruptions or failures of financial software systems can significantly affect organizations and their stakeholders economically and reputably and attract regulatory repercussions. In the last few years, the application of DevOps has become quite popular in the financial services industry. DevOps' concept helps in amalgamation of the development and operations of a company so that they can quickly and effectively deliver software products and services. Based on DevOps best practices, it is essential for an organization to attain flexibility, and quality, and to ensure the delivery of goods and services faster through the integration of the development and operation entity, automation of processes, and coming up with the development and delivery pipeline. New valuable features have been brought with the help of DevOps techniques, however, applying such approaches offers evidence that a better understanding of the behavior and functioning of the software systems are needed. This is where observability comes in to strengthen the situation. The term observe is a technique that is used with the aim of getting to observe how a certain system works, and its inner structure with the ultimate aim of being able to monitor, analyze, and improve the structure. The inclusion of observability into the

DevOps tradition in financial services technologies yields key benefits, including increased velocity of issue identification and remediation, better code quality, compliance, and end-to-end teamwork. However, all these also present organizational integration challenges, like security issues, information overload and having to adopt a transition in firm's culture.

Correlation between DevOps, Observability and Monitoring: -

Complementary principles like DevOps, monitoring, and observability are essential to today's development and operations procedures. Understanding this relationship as a whole is essential for building strong and effective software systems, especially in the financial industry where legal compliance, security, and dependability are critical. Monitoring is one of the most significant practices that needs to be carried out when working within the DevOps paradigm. However, what might be the single most important aspect of DevOps is its iteration and feedback, both of which are specifically driven by the analysis of data that is collected through monitoring. By integrating the monitoring into CI/CD pipeline, it is possible for the teams to collect and analyze system attributes, logs, and other metrics that are defined across the different phases of the software development life cycle. This makes it possible to find the causes of the problems at their inception, is effective in rectifying the problem, and can enable an organization to improve on the implementation of solutions that are already available. Observability makes the inferences in the field of DevOps less rigorous concerning the monitoring operations. Whereas this is about tracking the occurrence of certain issues that need management intervention, observability is a method used to examine and understand the issues as well as the patterns that lead to these specific issues. In other words, applying the concept of observability within DevOps enhances the ability of teams to understand the system's performance status, challenges and, if there are any pacemakers, identify the problems and optimize the system.

Integration Model for Observability with DevOps: -

When implemented side by side with DevOps, the use and adoption of specific observability practices become essential for companies in the financial services sector to deploy solid, efficient, and fully compliant systems. Therefore, this integration has to be carried out systematically in a way that accounts for the demands and issues of this industry.

1. ***Define observability requirements:*** When integrating observability with DevOps in the financial sector to ensure software development and operational resilience, the first step is to define the scope of the observability that is going to be utilized in measuring and revealing the state of the financial services applications and systems. This should be done in line with the organization's regulatory compliance requirements, critical performance issues, leverage target and overall workflow objectives as encapsulated by the organization's critical activities. For instance, in a trading platform used by a large investment bank, observability requirements might include:
   a) Measures concerning the time taken to execute trades, size of order books, latency of the market data feed, and degrees of usage of system resources. They are so useful when it comes to achieving the best in trading and to also pinpoint if there is an issue with capacity within a trading business.
   b) Logs capturing user transactional data, trade data, risk management data, and system data audits. These logs are helpful for compliance with the different rules and regulations belonging to different authorities, like Securities and Exchange Commission (SEC) and the Financial Industry Regulatory Authority (FINRA), which have standard rules for record keeping and audit trails.
   c) Open telemetry for reconstruction of intricate trade execution dependencies in multiple microservices like order routing, risk management and settlement services. These traces furnish full-fledged information regarding trade lifecycles, indicating potential problems or failures in the distributed system quickly.
   d) Therefore, by clearly specifying observability needs related to the nature of financial services applications, organizations can learn how to make the right calls when it comes to the points of observation and the data to be collected in order to remain operationally resilient.

2. ***Selecting the Right Observability Tools:*** Since applications and infrastructure in the financial services segment are intricate, and lots of data are produced in the observability framework, a tool alone is inadequate. Rather, organizations should use a set of observability tools, wherein the tools that different organizations will use are dependent on the type of need they have. These tools may include -

a) *Log management solutions:* Tools like Logstash or Elasticsearch or cloud solutions like AWS Cloud Watch logs or Splunk can help in pulling petabytes of log data from different sources.
b) *Distributed tracing tools:* Jaeger, Zipkin or similar or AWS X-Ray, can aid in distributed tracing, which provides insight on how the requests go through the MS and where the slow or failed requests are likely to come from.
c) *Application Performance Monitoring (APM) solutions:* APM tools, such as 'AppDynamics', 'Dynatrace', or 'New Relic' that work at the application work level and the code level can monitor metrics, behavior and traces of an application to help in the identification of performance issues easily.
d) *Infrastructure monitoring tools:* Some of them are Prometheus, Datadog, or Azure Monitor, where metrics and logs of sub tier components of the technology stack, mostly servers, databases, and networks, are collected for top-down system level observability.

The choice of observability tools can also be narrowed by certain potentially valuable characteristics, such as the products' ability to integrate with other systems and their compliance with standard protocols of industries the business is active in, as well as the question of whether the products are scalable, such as the incorporation of the Financial Information eXchange (FIX) protocol, data visualization tools, analytics, security, and data privacy compliance.

3. ***Integrate with DevOps toolchain:*** Like with any other tool that generates data, to get the most value from observability data, it needs to fit seamlessly into the current DevOps toolchain. This makes it possible that observability data follow the program from the time it is coded, tested and deployed to when it is run in a production environment. For example, observability tools can be integrated with –

a) *Continuous Integration/Continuous Deployment (CI/CD) Platforms:* It is further ideal for developers to incorporate observability along with CI CD tools like Jenkins, GitLab, and Azure Pipelines to collect and visualize observability data for build-test-deploy phase to be able to determine where the problems lie closely and fix them with speed.
b) *Issue tracking systems:* Links with other tools like Jira, Azure DevOps Boards, or GitHub Issues make it possible and easy to build or monitor issues directly according to the observability data insights and ensure that the operations and development teams are in sync.
c) *Collaboration tools:* Real-time notifications and alerts are enabled by the integration of observability data with collaboration platforms like Slack, Microsoft Teams, or PagerDuty. This expedites the process of responding to and resolving incidents.

It is crucial to note that financial services organizations should adopt observability as a DevOps practice to address the issues of the separated development and operations teams. This will inform all team members about the system's behavior so that they can collectively work on bringing about changes and improvements.

4. ***Establish alerting and notification strategies:*** Observability data is of most use when it identifies and triggers the resolution of potential problems before they manifest themselves in their negative effects on customers or the business at large. Thus, depending on the observability data, financial services organizations should set the alerting thresholds and notification processes firmly. An alerting strategy for a trading platform, for example, might involve –

a) Applying time thresholds for the execution of trades with the help of historical data and the requirements of the company. When the execution time of the trades is beyond these thresholds, the alarms can be raised with the relevant groups to perform analysis and deal with relevant issues, if any.
b) Setting up alerts regarding the latency of the market data feed because possible delays in this type of data may put traders in a compromising position when it comes to making decisions on the stock to buy and sell, or in case they need to avoid certain securities, thus leading to incurring a loss.
c) Setting up alarms for security activities, for instance, extraneous access attempts or suspicious user activity using logs. These alerts can be forwarded to the security group for further examination and action to be taken.
d) Designation of the notification channels and procedures depends on the categorization of the problem and its possible consequences. Some critical alerts can notify the on-call engineers, while others might go to the monitoring dashboards or ticketing systems. Alerting/notification can effectively solve the problem before it arises, reducing losses, business downtime, and damage to reputation.

5. ***Promote continuous improvement:*** As with most organizational practices, the use of observability within DevOps is a continuous process, thus requires constant finetuning based on feedback coming from development, operations, and clients. While implementing financial services systems, new regulatory requirements or business needs may arise, and as a result of these changes, observability must catch up to the changes so that the collected data is useful.

   To promote continuous improvement, financial services organizations should:

   a) Ensure the interactions so critical for development, operations, and business teams are effectively communicated and executed. Forums or assemblies, whether global or per functionality, can be crucial to receiving opinions about the data obtained through observability, as well as precisely observing where refinement might be needed or where observability adheres to business change.
   b) Regularly update the decision-makers on changes in the observability need, data feeds, and alert generation techniques. New services or features that are rolled out should prompt changes to the observability practices to incorporate the data that is needed as well as proper alert generation.
   c) Support exchange of information and knowledge about the observability tools and practices, as well as training on the tools. There is no one-size-fits-all fix for healthy culture, and it could require constant reinforcement, but providing regular training and documentation could be beneficial to continuously remind teams to be good at using the observability data and tools.
   d) Taking that into consideration, the analysis provides insights on how to make use of the observability data and apply them to enhance processes and make optimizations. For instance, defining frequent performance issues or failure trends increases the chances of rectifying, redesigning or optimizing the application. Focusing on the improvement of the observability data as the feedback loop enables financial services organizations to sustain operational resilience and optimize system performance while aligning with everchanging regulations and business requirements.

Benefits of Integrating Observability with DevOps

a) ***Faster incident detection and resolution****:* Typically, metrics and trends are not only provided to specific thresholds but also to preconfigured alarm systems, which may appear insufficient when it comes to large-scale distributed systems. While compared to observability, metrics provide a direct window into the system and are easier to understand, observability unifies metrics, log data and distributed tracing. This broad plan and action help to focus on problem search and diagnosis and, therefore, accelerate the resolution of such problems. For instance, think of the case of an organization that is in the financial sector, and has a trading floor where its personnel trade securities. The problem with traditional monitoring is that it does not allow the identification of the source of the problem, which can be in any component or dependency. On the other hand, by using observability data, including distributed tracing, the developers and operation teams will easily point to the particular service or component that is most likely to be the cause of the bottleneck so that adequate measures can be employed to rectify the situation. The benefit of faster incident resolution is that it could lead to fewer impacts on business and, thereby, a lesser amount of revenue lost. Time is a key factor in the financial services industry, so the capability to address issues sustainably can serve as a major advantage in retaining customers' trust and be less of a disadvantage in terms of revenues lost.

b) ***Improved software quality:*** Observability helps to continuously monitor the system after, during, and before the code is deployed in various steps such as development, testing, and production. The observability data can be gathered and analyzed during the development and testing of the software so that issues such as bugs, potential performance problems and bottlenecks may be ironed out prior to the software being rolled out to production. Such measures can be applied to ensure that financial software offered to the public will provide the best quality, security and firmness since financial data is sensitive. Thus, the financial institutions can reduce the potential for costly shocks, decrease the time that the systems are out of order, and satisfy the customer by releasing higher quality software.

c) ***Enhanced regulatory compliance:*** This segment involves various rules and regulations covering the field, like companies, the Securities and Exchange Commission (SEC), the Financial Industry Regulatory Authority (FINRA), and the Basel Committee on Banking Supervision. Aiding to these regulations attracts severe penal consequences in terms of fines, legal procedures, and reputation. Observability data becomes extremely valuable in meeting auditing and reporting criteria. For instance, the regulation from the US Securities and Exchange Commission known as Regulation Systems Compliance and Integrity (Reg SCI) demands that financial institutions have strict measures for the operational continuity of their systems. Observability data could allow an organization to meet the elements of Reg SCI regarding risk management, incident reporting, and systemic testing of the systems.

d) ***Streamlined collaboration:*** Observability is beneficial to DevOps teams as it helps them to discuss and work with a mutual understanding of application behavior. When decision makers from different parts of the organization are presented with the same observability data, they are in a position to solve observed problems more efficiently, find the causative factors to problems more efficiently, and come up with solutions to the observed issues more efficiently. This integrated approach also minimizes mysteries and fosters DevOps, the practice that aims at everyone’s responsibility in creating quality software and high-performing systems. Development and operations are two sides of one coin and, when combined in the most efficient manner, are capable of increasing the rate of solving incidents, making changes more smoothly, and providing a higher value to customers.

Challenges and Considerations

a) ***Security concerns:*** Financial services include the operation of customers’ information, such as their identity details and financial status. Therefore, handling observability data has to be done with a lot of caution in regard to their storage, collection, and access [28]. Observability’s implementations within financial services must ensure that data in transit and at rest is encrypted, that effective and proper access control grant mechanisms are in place, and that data is anonymized. In the same respect, there should be security audits at least once a year, along with security assessments for potential risks.

b) ***Data overload:*** There are three things that an observability system produces, and they are events or logs, performance metrics, and distributed traces. However, failing to screen and rank such huge information streams appropriately can become a problem since useful and relevant information may be lost in the flow of large amounts of information, and potential inefficiencies and even issues can be left unnoticed. In order to meet this challenge, it is necessary for financial institutions to consider applying approaches to the selection and organization of key observability data. It is possible to use approaches like the logs’ correlation, anomalies, and metrics’ grouping to pay only attention to significant data.

c) ***Cultural shift:*** It is common that the implementation of observability is aligned with DevOps methodology, and this change usually takes some time at the organizational level. The current approach of reacting to issues as they arise has to be replaced by continuous monitoring, which is supported by observability. Really encouraging the reactivity of the observability into teams, DevOps means raising awareness among every member of the team of the added values of the observability, the professional development of the tools of the observability, and the constant evolution of the mindset of the observability. All in all, this became a generational shift, which can be seen as both a weakness and a strength when seeking to utilize all the potential of observability in financial services technologies.

Future Trends

a) ***Artificial Intelligence (AI) and Machine Learning (ML):*** Observability and AI and machine learning what may have been the case even a year ago has since changed drastically. AI/ML tools can be applied to different processes dealing with observability, from the root cause analysis of problems to the prediction of equipment failures and incident solving. For instance, supervised machine learning can be used on recorded observability to predict likely problems that may occur in the future in order to prevent worst-case scenarios. These models can also suggest the possible measures that should be taken to correct the problem and thus facilitate the solving of the incident. Besides, using the observability data, AI/ML can be used to predict hardware or software failures and prevent them from occurring, thus reducing system downtime.

b) ***Security considerations for observability in financial services:*** As observability practice deployment comes into light in the financial services industry, organizations must guarantee the security measures of acquired observability data. The openness of financial data, coupled with the nature of observability, which offers a great deal of information in comparison to traditional approaches, requires a global approach to security. Security features are also important, with attention paid to the encryption of data both in transit and at rest in order for observability to be implemented. Financial institutions should also ensure the observability of data privacy through encryption via standard security protocols. Another key component is access control systems, which must restrict the availability of observability data to employees who need it for their work. Implementing RBAC and multi-factor authentication can avoid or minimize the chances of an incident such as firewall intrusion or leakage of employees' databases.

c) ***Data minimization*** is another factor whereby it is required that financial institutions only acquire and retain the observability data that is relevant for the use cases of that institution. This eradicates the chance of data leakage and helps in adherence to the set information technology data privacy provisions. This operational reality suggests that there should be frequent systematic security reviews and reporting to determine and fix any existing gaps in the use and deployment of observability in organizations. It is important that such audits encompass all the data collection processes, storage procedures, ways of accessing and analyses of the observability infrastructure. VI.

Conclusion

Applying observability in synergy with DevOps methodologies for financial services technologies leads to multiple advantages, including swift identification of issues and their resolution, improved applications' quality, compliance with regulations, and improved cooperation between the development and operational departments. Thus, understanding the specifics of their software systems' behavior enables financial institutions to prevent certain problems, reduce service interruptions, or provide a high-quality customer experience. Observability transforms monitoring into a proactive process that continues throughout the organization's operations, allowing organizations to be prepared for various problems and maintain operational readiness. However, the process of attesting observability with DevOps practices in the financial service domain also comes with challenges such as security compliance issues and data overload problems that arise from a paradigm shift in the organizational culture. To manage these challenges, there is a need to adopt a multifaceted approach that embraces proper security measures, an efficient manner of handling data, and a culture of consistent learning and development. With the advancements in AI and ML in place, observability automation has the potential to be used for functions as simple as root cause analysis, predictive upkeep/repair, and incident diagnosis, among others. These technologies can complement the benefit that observability data brings to financial institutions in the sense that it can provide deeper and more proactive optimization of their operations. Comprehending the requirements for constructing and managing sound, resilient and secure financial services technologies in the era of rapid innovation, observability is an indispensable player. Hence, when financial institutions adopt observability and bring it together with DevOps, they will be able to contest and meet the regulatory requirements while at the same time offering commendable customer service and thus be a forerunner in the market.

Reference: -

1. C. Longbottom, "The pros and cons of CI/CD pipelines," SearchSoftwareQuality, Apr. 22, 2021. https://www.techtarget.com/searchsoftwarequality/tip/The-pros-andcons-of-CI-CD-pipelines
2. I. Jada and T. O. Mayayise, "The impact of artificial intelligence on organisational cyber security: An outcome of a systematic literature review," Data and Information Management, pp. 100063–100063, Dec. 2023, doi: https://doi.org/10.1016/j.dim.2023.100063.
3. J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," Journal of Computer and System Sciences, vol. 80, no. 5, pp. 973–993, 2019, doi: https://doi.org/10.1016/j.jcss.2014.02.005.
4. L. Rivers, "Strategies for Reducing the Risk of Data Breach Within the Strategies for Reducing the Risk of Data Breach Within the Internet Cloud Internet Cloud," 2020. Available: https://scholarworks.waldenu.edu/cgi/viewcontent.cgi?article=11074&c ontext=dissertations
5. J. R. Garcia, "SOC Analyst Level 2: TryHackMe: Log Analysis: Intro to Logs," Medium, Oct. 19, 2023. https://medium.com/@joseruizsec/socanalyst-level-2-tryhackme-log-analysis-intro-to-logs-b7b2bfbc66b5
6. L. Vishnoi, "Top 10 Observability Trends in 2024," Middleware, 2024. https://middleware.io/blog/top-10-observability-trends-in-2024/ (accessed May 22, 2024).
7. Samuel Onimisi Dawodu, Adedolapo Omotosho, Odunayo Josephine Akindote, Abimbola Oluwatoyin Adegbite, and Sarah Kuzankah Ewuga, "CYBERSECURITY RISK ASSESSMENT IN BANKING: METHODOLOGIES

AND BEST PRACTICES," Computer science & IT research journal, vol. 4, no. 3, pp. 220–243, Dec. 2023, doi: https://doi.org/10.51594/csitrj.v4i3.659.

8. "A. Mahida, "Automated Root Cause Analysis with Observability Data - A Comprehensive Review". Journal of Engineering and Applied Sciences Technology. Volume 5(6), pp. 2-4, 2023. DOI: doi.org/10.47363/JEAST/2023(5)230"
9. A. Mahida, "A Review on Continuous Integration and Continuous Deployment (CI/CD) for Machine Learning," International journal of science and research, vol. 10, no. 3, pp. 1967–1970, Mar. 2021, doi: https://doi.org/10.21275/sr24314131827.