International Journal of Research in Engineering, Science and Management
Volume 5, Issue 6, June 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

DevOps KPI Challenges – Continuous Monitoring Safeguards

Rakhi Parashar*
Technology Architect Delivery Team Lead, Learning and Management Delivery, Accenture, Mumbai, India

***Abstract*: Along with Continuous Integration and Continuous Deployment, Continuous Monitoring is a key DevOps strategy. Continuous Monitoring (CM) is also an important part of the Agile and DevOps processes. We live in a digital world where speed always wins. To keep up with client demand and expectations while remaining competitive in the market, each Endeavor must move faster and more flexibly in application development, testing, QA, and monitoring. As a result, monitoring the tasks completed by their DevOps/Infrastructure teams is required, as is reducing manual processes and automating the majority of them. This new manner of working has the potential to improve Operative's productivity while also bringing new profits and client satisfaction. DevOps is used to respond faster and more effectively to continuously changing consumer needs. The continuous delivery and continuous deployment model is used by DevOps. Continuous Monitoring is an important (and sometimes overlooked) part of our DevOps lifecycle, as it keeps an eye on our Continuous Delivery and Deployment systems in any organization. The importance of continuous monitoring, DevOps KPIs, and tools for the same are discussed in this paper.**

***Keywords*: Continuous Monitoring (ConMon), Continuous Integration and Continuous Deployment (CICD), DevOps, SDLC.**

1. Introduction

Speed wins hearts in this Digitized and it has become a Uprising, each and every innovativeness needs to move quicker and more compliantly in application development, testing, QA and Monitoring to keep up with the Client's demand and hope and be competitive in the market. Fixing issues/bugs for applications, servers and application running on them manually became difficult. Monitoring tools are helping to automate these tedious manual tasks. Along with CICD, Continuous Monitoring is also one of the disciplines of DevOps and implemented as a practice by agile and DevOps projects. The benefits of this approach include reduced time, risk, and expense of software delivery. This paper looks at the benefits of CI and CD together and the requirements for effectively deploying these practices in IT organization.

2. Literature Review

Why Continuous Monitoring?

DevOps KPIs can help teams track different events in the process and guarantee continuous improvements. Implementing and tracking the right system of measurement/metric is not always simple:

1. *So many KPIs:* Teams face the problem of plenty as they try to find out the right metrics for evaluating DevOps success. So many metrics often lead to information overload.
2. *Unproductive Analysis:* DevOps KPI dashboards may not always convey a story and teams can struggle to correlate and extract actionable intelligence from their data.
3. *Absence of Flexibility:* Teams lack flexibility in updating or defining new metrics. The easily measurable metrics may not always be the most helpful ones.

Fig. 1.

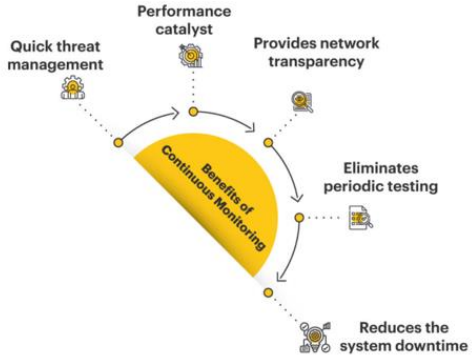
Performance catalyst
Quick threat management
Provides network transparency
Benefits of Continuous Monitoring
Eliminates periodic testing
Reduces the system downtime

Fig. 2.

Create DevOps Metrics Dashboard with tools:

Monitoring tools help you stay on top of your DevOps activities and get actionable intelligence quickly with a ready-

*Corresponding author: rakhi.parashar@accenture.com to-use and customizable DevOps KPI dashboard.

- *Ready Integration:* Use out-of-the-box connectors to unify data across tools and get end-to-end pipeline visibility.
- *Visual Dashboard:* Use pre-built apps and templates to set up your DevOps dashboard and visualize key performance indicators.
- *Industry Standard Metrics:* Quickly implement and track the four key DORA metrics (Deployment Frequency, Lead Time for Changes, Change Failure Rate, and Time to Restore Service) to ensure DevOps success.

Safeguard Continuous DevOps Monitoring:

Track metrics in real time and proactively address performance issues to continually improve end-user experiences.

- *Custom Metrics:* You can easily define your own metrics to track progress and make informed decisions.
- *Traceability:* Seamlessly trace issues across the DevOps lifecycle to identify bottlenecks and troubleshoot issues.
- *Upbeat Response:* Enable faster feedback loops to continuously improve user experience

Solution Benefits:

Ensure DevOps success with real-time visibility into CI/CD pipelines using industry-standard and custom metrics.

Software Engineering Fast-track time-to-market for new features and functionalities. Track metrics including Development Frequency, Speed of Deployment, and Mean Lead Time for Changes. Quality Assurance - Identify defects at an early stage. Track metrics such as Change Failure Rate, Security Test Pass Rate and Average Test Code Coverage to continually improve release quality.

Operations -Track system reliability and maintainability metrics such as Mean Time to Resolution (MTTR), and slice and dice data to identify and resolve breakdowns faster.

Measuring DevOps Success:

While most organizations have embraced DevOps developing the right culture, establishing communication channels, and implementing advanced tools, they are still unable to assess the success of their DevOps initiatives. DevOps KPI measurements can help organizations in this direction. If selected carefully, DevOps KPIs can offer a holistic view of the impact and business value of DevOps initiatives. Organizations should start with defining their measurement goals before they start looking out for DevOps metrics dashboarding tools and solutions. Business and development teams need to agree on the right set of metrics while considering which business outcomes have the maximum impact on ROI and user experience. Once organizations finalize their metrics, they should evaluate the time and effort involved in gathering metrics from their existing DevOps setup. Every time a new tool is added to the DevOps setup, the team should evaluate its impact on time-to-insight. Solutions offering easy integration with DevOps tools along with self-service analytics can help organizations add agility to their DevOps KPI measurements and monitoring.

How can DevOps KPIs help in gauging business value?

Organizations can track business value over a period based on certain focused and custom KPIs. For instance, the metric ‘Innovation’ can be defined as a function of every feature enhancement delivered to customers, excluding the bugs. Tracking this metric can help businesses determine how much they have actually worked towards improving their product or service, against removing its existing, lingering flaws. Similarly, Mean Time to Resolution (MTTR) can provide insights into customer satisfaction and Velocity can help in assessing how much work the team is capable of doing in a given time. Such metrics and their trends over a period can offer organizations data-driven predictability and can be used as a business driver.

DevOps continues to gain traction among organizations as demand grows for digital product and platform development. According to Gartner, 87% of business leaders believe digitalization is a priority. Every DevOps transformation requires a dedicated, continuous learning process and effective implementation to reach maturity. If a practice or pattern is passed over or ignored, it can put a damper on DevOps success. Continuous monitoring, for example, is a very important part of every DevOps life cycle that is often overlooked.

Understanding Continuous Monitoring: The practice of continuous monitoring helps to collect and analyze outcomes, statuses, exceptions and key metrics within each step of the DevOps process – from development to deployment and production. Additionally, it offers deep insight for DevSecOps teams by enabling compliance triggers and security alerts configuration. It allows the organization to detect issues or security concerns throughout every phase of the DevOps lifecycle.

What should be observed?

An effective continuous monitoring process collects enough data to gain sufficient understanding of both successes and failures of the organization’s systems and processes. Some of the events collected by the continuous monitoring systems will require immediate attention like system logs, Memory, CPU utilization, request/ system response time, failure rate, Latencies, uptime, access time, etc.…

A combination of technology and strategy helps ensure that the right data is collected at the right time. The next step is finding the balance between monitoring and analysis.

How to Achieve Balance?

One of the most difficult decisions in log monitoring and analysis is to strike a balance. If you collect too little information, you risk missing early warning signs or vital notifications. When you collect too much information and become engulfed in analytical paralysis, you run the risk of overlooking important details. To strike the correct balance, you must first grasp the system's continuous monitoring processing capacity and make full use of it. This may also include data preparation activities during intake to guarantee that the collected data is processed and stored efficiently. Without regular monitoring of the DevOps life cycle, a DevOps transformation is unlikely to reach full maturity.

Technology is always improving, and teams must release software upgrades to the production environment as rapidly as possible to stay ahead of the competition. Organizations are turning to DevOps to obtain a faster reaction to changing client needs. Monitoring each level of the DevOps pipeline and identifying any compliance or security issues that may obstruct the development of high-quality output is crucial. Continuous Monitoring in DevOps is in charge of keeping an eye on all stages of the pipeline and always monitoring any potential dangers.

Continuous Monitoring in DevOps is an automated technique that assists DevOps teams in detecting compliance concerns at various phases of the DevOps process. As the number of cloud-based applications rises, the IT security team will need to implement a variety of security software solutions in order to mitigate security threats while ensuring privacy and security. Continuous Control Monitoring is another name for continuous monitoring in DevOps (CCM). It doesn't simply apply to DevOps, but to any area that demands attention. It gives sufficient data for decision-making by allowing for easy tracking and speedy error detection. It offers feedback when things go wrong, allowing teams to assess and take corrective action in a timely manner. It's simple to do with the right tools.

Goals of Continuous Monitoring in DevOps:

At the time of the production release of the software product, Continuous Monitoring notifies the Quality analysts about any concerns arising in the production environment.

Continuous Monitoring in DevOps helps organizations track the operational performance of the app. It supports monitoring the user’s behavior at the time of new application updates.

It helps teams understand the impact of the recent updates, the real-time data on the user interactions, and the overall user experience. This data is helpful in the root-cause analysis of the situation and the fitness of the IT infrastructure, offsite networks, and deployed software.

Types of Continuous Monitoring in DevOps:

Infrastructure monitoring: Under this, the IT infrastructure of the organization responsible for delivering the end product is monitored using DevOps monitoring tools. This infrastructure includes the software, hardware, servers, data centers, networks, etc. It gathers data from different IT systems and analyzes that data so that the decisions to improve the product or service are made easy.

Application monitoring: This type helps in monitoring the performance of the released application. It measures uptime, time taken in completing a transaction, system responses, API responses, servers, and UI sides of the system.

Network monitoring: Network monitoring aims to detect and mitigate all network-related issues and notify the respective team to prevent crashes. It provides the status of firewalls, routers, switches, virtual machines, etc., and their functioning.

Advantages of Continuous Monitoring in DevOps:

Continuous Monitoring puts IT firms ahead of their competitors that still undertake batch analysis on their data. Continuous monitoring aids in the gathering of crucial data about IT infrastructure, applications, and networks. It allows for real-time monitoring of the organization's critical data. Let's have a look at the advantages:

- *Network transparency:* DevOps delivers comprehensive transparency on the status of the technical setup through continuous monitoring. It deciphers the system, collects and analyses critical data on the fly, and guarantees that vital trends/events are not overlooked due to the system's ambiguous indicators.
- *Quick incident response:* Continuous monitoring shortens the time between discovering an issue and notifying the response team. The ability to respond quickly to such situations reduces the likelihood of operational problems and security threats. Consistent system monitoring offers an alarm mechanism and real-time security monitoring to limit or avert harm that may otherwise cause application performance concerns.
- *Reduced system downtime:* The primary goal of Continuous Monitoring is to keep the system operational and error-free. It can be accomplished by responding quickly to app performance issues before they cause system downtime and service interruptions that affect end users.
- *Business performance catalyst:* With all of the advantages of CCM, Continuous Monitoring alleviates the load of dealing with app issues that negatively impact the customer experience and saves the company from losses. Continuous Monitoring systems help QA, development, sales, marketing, and customer service teams make business choices by providing crucial user and system data.

Best Practices for Continuous Monitoring in DevOps:

IT ecosystem has its key areas to track like user behavior, server health, app performance, development targets, and system strengths and weaknesses, etc. Below are some best practices which we can focus while performing Continuous Monitoring:

- *Infrastructure monitoring* systems must keep an eye on server and database health, storage, response time, security, user permissions, networks, and performance trends, among other things. Server Spec, Sensu, and so on.
- *Network monitoring* tools must keep track of network latency, server bandwidth, network packet transfers, multi-port metrics, and other factors.
- *Application monitoring:* User response time, user interactions, page loading speed, third-party application speed, browser speed, SLA status, and so on are all things that application tools must keep track

of. ELK, EFK, Prometheus, Grafana, CAdvisor, AppDynamics, and more similar technologies are available.

Continuous Monitoring in the DevOps Environment:

Technology teams may monitor system anomalies and use metrics to take corrective action with the help of robust and diverse solutions.

- *Conduct a risk analysis:* Businesses must recognize the value of risk management. Its job is to find high-risk locations. High-risk assets, for example, require more security controls.
- *Select a security control system:* The second step's risk analysis will offer the DevOps team with enough information to determine which areas require additional attention. To safeguard the system, IT staff can use passwords, firewalls, antivirus, encryption, and other security controls.
- *Set up monitoring tools:* As businesses set up Continuous Monitoring technologies, the monitoring tools begin to collect crucial security control data. Log files from the deployed programme are captured by the monitoring tools. These log files store information about all application activities and interactions, such as security threats and other operational parameters.
- *Evaluate data:* In the end, it is the data that is examined to produce relevant insights that aid in future business decisions.

We all know that continual monitoring takes a lot of time and effort. It aids testers in gaining a thorough understanding of errors, which aids quality analysts in their testing efforts. Organizations must release adequately tested software products, i.e., in the real world, to effectively manage continuous control systems. The test results will not be accurate if the software is tested using emulators and simulators. As a result, they must be tested in a real-world setting to ensure accurate results in both human and automated testing.

3. Conclusion

In this research path of continuous Monitoring, we are highlighting popular CM tools like Prometheus, Grafana, cAdvisor, Elasticsearch Logstash Kibana (ELK stack), Elasticsearch Fluentd Kibana (EFK stack), AppDynamics, various Cloud Services for Monitoring (AWS, Azure, Google, etc..), Linkerd, DataDog, etc. Please refer the table structure for few of the Monitoring tool reference. With the changing climate, spotting difficulties early on ensures that you keep one step ahead of the game by resolving the issue before it becomes a severe problem. It allows businesses to keep track of application performance, infrastructure, and network traffic in real time. It constantly monitors how users engage with any new feature of the application. Sensu, PagerDuty, Slack, Ansible, Chef, Puppet, and other Continuous Monitoring solutions make Continuous Monitoring easier by sharing insights into the application's performance and productivity, as well as notifying early symptoms of faults. Continuous monitoring in DevOps is effective throughout the SDLC and even after deployment. We can also automate database performance monitoring, which is extremely useful when querying data for analysis. It allows businesses to track application performance, infrastructure, and network traffic in real time. It maintains a close eye on how users behave when interacting with any new feature of the application. It also gives firms information on how capable their IT infrastructure is to deal with such problems. Sensu, Serverspec, PagerDuty, Slack, and other Continuous Monitoring solutions make Continuous Monitoring easier by sharing insights into the application's performance and productivity, as well as detecting early symptoms of faults. Continuous monitoring is used in DevOps from the start to the finish of the SDLC, and even after deployment. Automating database performance monitoring is critical to a successful DevOps implementation.

References

[1] http://klera.io/use-case/devops-kpis/
[2] https://www.browserstack.com/guide/continuous-monitoring-in-devops
[3] https://dzone.com/articles/role-of-continuous-monitoring-in-devops-pipeline
[4] https://medium.com/devopscurry/understanding-continuous-monitoring-in-devops-f6695b004e3b
[5] https://www.openxcell.com/blog/continuous-monitoring-everything-you-need-to-know/