OverOps

AN OVEROPS TECH BRIEF

Continuous Reliability

The problem with CI/CD

Continuous Integration/Continuous Delivery/Continuous Deployment (CI/CD) introduces automation into the software release cycle that allow organizations to release software more consistently and more quickly. These processes require a certain level of automated testing for each function as code gets promoted through the various stages. In fact, these concepts are predicated on the ability to apply stringent testing to applications as part of the automation process. Organizations that employ this level of automation are typically very concerned about quality, but the speed this introduces can also affect the overall functional quality of an application as a whole. They want code to be safe to promote but still know that issues will get through. It is difficult as the level of automated testing that is required scales exponentially with the amount of automation.

Organizations try to evaluate overall quality of an application in each deployment environment by analyzing their log files but this provides only limited insight into raw numbers and not enough detail into which function or better yet, why something is failing. They provide no insight into new or reintroduced errors. Understanding frequency or failure rate is impossible. CI/CD automation is predicated on delivering quality software but it is a challenge to understand this using log files alone.

Further, testing frameworks are still fairly manual in that they are defined by humans and it is impossible to consider all the corner functional cases or the permutations of data that might need to be tested to get complete coverage. Actually, 100% test coverage is 100% impossible.

Introducing Continuous Reliability with overOps

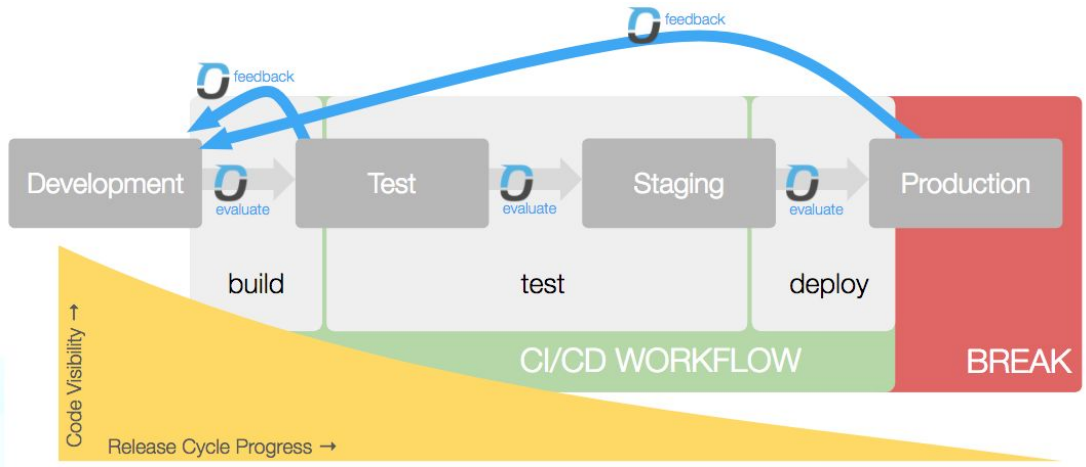
OverOps capture complete state of the JVM every time an error is logged or an exception is thrown. This allows us to gain deep insight into each individual event. We can also compile this information across an entire application, library, class or deployment or any other arbitrary boundary to provide deep insight into the overall functional quality across the code. Our unique set of data allows us to identify both know and unknown errors and we can also

classify events, providing information on whether they are new, reintroduced and their frequency and failure rate.

- **Quality Gates:**
  This information provides deep insight into your application that can be used in your CI/CD workflow to evaluate the overall quality of a release and help you evaluate whether or not it is "safe" to promote code. You can aggregate all of the issue data for a deployment or release to determine a very granular understanding of the quality of the application and use this to regulate and decide when a release is safe to promote. OverOps publishes this data via API so that you can view all critical issues or regressions associated with a release directly in your CI/CD tooling (ie Jenkins) so you can incorporate this into your workflows.

- **Feedback Loops**
  Using OverOps data to inform the testing habits of your developers greatly increase their productivity and can be a part of their common workflow and testing in continuous integration. The information feedback to them will help the not only troubleshoot, but to write better automated test scripts and to use real production data that caused issues as a basis for testing in lower environments. OverOps can also help you understand version numbers of all your code and provide a window into the interactions of functions, classes and services, so you can determine where to focus remediation efforts when there is an issue.

Typically, teams use OverOps in the context of a CI/CD workflow to help govern the release and to provide a valuable feedback loop.



About OverOps

OverOps provides net new machine data from applications and services to help organizations effectively evaluate the reliability of their software and implement a culture of accountability.

OverOps developed a unique approach to gathering machine data and it changes the way we think about log files and how we use them to both troubleshoot and to derive the overall quality of an application or service. OverOps combines static and dynamic analysis to collect complete contextual data for every error and exception thrown –both caught and uncaught– in any environment, including production with minimal performance impact, securely and without any requirement to modify code. It is code-aware and delivers net new and structured machine data that provides granular detail about every error, its related application and the environment in which it was found

This deep visibility into the functional quality of applications and services not only helps developers more effectively troubleshoot, but also empowers DevOps to build metrics dashboards, implement continuous reliability and enhance the entire software delivery supply chain. As more organizations aim to innovate faster and deliver a seamless digital experience for their customers, OverOps helps avoid costly downtime that can lead to lost revenue and brand degradation.