

OverOps & GitHub

Get Deep, Code-Level Context to Quickly Resolve Any Error

OverOps now integrates with GitHub code repositories, providing support for git blame and automated source attach

The screenshot displays the OverOps dashboard. At the top, it shows 'OverOps Prod' and 'OverOps is active' on '21 JVMs on 9 machines'. A 'Log Error Anomaly in BatchReduce' is highlighted, with a graph showing a spike in error frequency around July 2nd. Below the graph, a list of error logs is shown, with 'CommonTablePartitioner.execute' selected. The main view shows the source code for 'SqlBatchReducer.execute' with a git blame overlay indicating the last author was Julia Rogers. A 'Recorded Variables' panel on the right shows the state of the application at the time of the error, including 'Thread-local state', 'Log message', and various application parameters.

Promote a culture of developer accountability

As companies accelerate their time-to-market and increase their release velocity, more and more software defects tend to find their way to production. As a result, software engineering teams often spend up to 60% of their time on finding and fixing errors from previous release, pulling resources from the next project.

Integrating OverOps together with your code repository encourages developers to take ownership and build a culture of developer accountability. Empowering developers with code-level visibility across the entire pipeline.

See who was the last author who changed the code across the call stack of each error

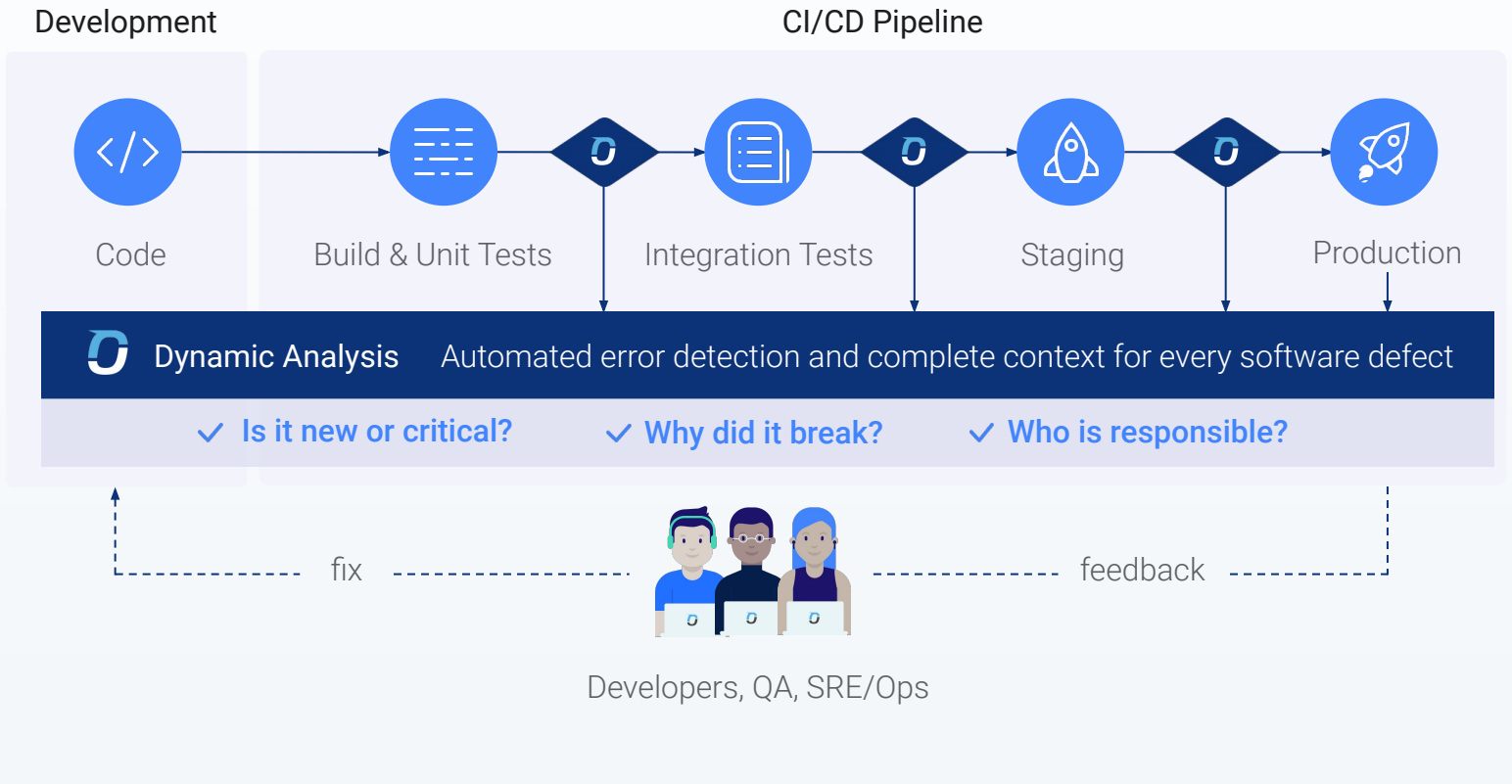
Link errors to commits and view the latest code changes in your git repository

Assign every new and critical issue to the developer who is responsible for fixing it

Capture detailed error snapshots with the source code and variable state for every error

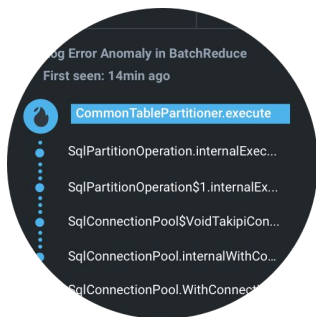
Continuous Reliability. Delivered.

OverOps analyzes code at runtime to deliver application error analytics that help engineering teams identify and resolve critical application errors.

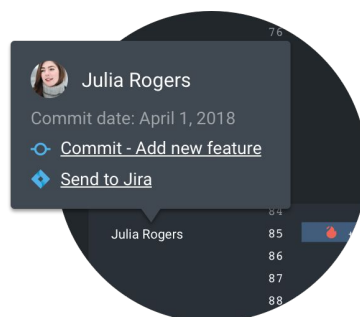


Know When, Where, and Why Code Breaks in Test & Production

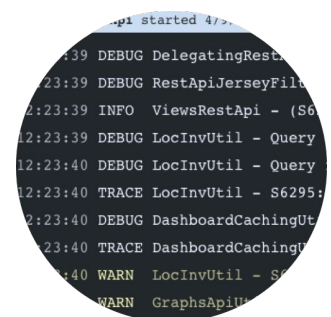
With error snapshots from OverOps



See the stack trace, source code and variable state for every error



Assign new issues to the right developer, at the right time, with the right data



Capture the relevant TRACE & DEBUG level logs, and the state of the host or container