Customer story / Media

How Fox Manages Creative Assets Without Wasting Time on Errors

OverOps Helps to Ensure that Film Development at Fox Isn't Slowed Down by Application Errors

FOX

Fox Entertainment Group is one of the biggest names in entertainment today, offering filmed entertainment, television stations, television broadcast networks, and cable network programming. OverOps is currently monitoring the FOX MediaCloud, the company's creative assets management ecosystem which evolved from FOX Esprit and supports 12 divisions across Fox.

12K+	1.8	$13.2B	50
Employees	Billion Subscribers	Revenue	Languages Supported

Production Monitoring Ecosystem: OverOps, New Relic, splunk>, slack

Highlights

- OverOps helps Fox maintain an error-free digital management system for its film entertainment group.
- OverOps helps Fox address errors proactively so that they fix them as soon as they occur.
- OverOps reveals the full variable state so that Fox can focus on immediately fixing each problem rather than sifting through logs.
- With OverOps, Fox's average error resolution cycle was cut down by 98%, from 2 days to 15 minutes.

Key challenges and pain points

When we used APM tools without OverOps, we didn't have enough details about errors, which forced us to go sift through the logs. Based on the error message, we'd identify the frequency. Then we had to attach a debugger to the environment to actually catch the root cause of the error.

OverOps has taken over that entire process, proactively instead of reactively.

"OverOps is the time-saver we've been looking for in other tools, that gives detailed analysis without having people access servers directly."

Tel: 1-800-324-3053 www.overops.com
797 Bryant Street, San Francisco, CA 94107

Example problem that OverOps helped resolve:

We saw immediate value from OverOps. During our trial we actually had a scenario where one of the developers had spent probably a day and a half, almost two days, trying to identify the root cause of an issue.
Then, we just happened to instrument with OverOps in that environment and within probably fifteen minutes, we saw the error and saw what was causing it.
It was a numeric exception error. It was because a variable had a slash in it instead of just pure numbers.
We have a lot of workflows that sometimes cause an error, meaning we had to go through the logs in search for the cause. After we found it, we had to reproduce in order to solve the error.
In one case, it was a permission exception.

OverOps gave us the exact variables and the user account in which the error had happened, allowing us to identify the problem and see that the user was not properly configured in the system, which is why these exceptions were being thrown.
Now that OverOps gives us the exact conditions behind each error, we can see why it happened and if it's critical for us. So we can catch the issue in action, as opposed to having developers search through logs or attach debuggers to the production or pre-production environment. That's the time-saver we've been looking for in other tools, that gives detailed analysis without having people access servers directly.

How are you integrating OverOps with your daily workflow?

We're using it three ways. One is obviously the dashboard, we periodically go into the dashboard to follow up with new errors. We also have email alerts that get sent out on a daily basis, and we recently instrumented a Slack channel to deliver the alerts on an hourly basis. With slack, developers can look at performance hour by hour and see if there is something new that pops up.
Before OverOps, we sometimes had to spend two days trying to identify the root cause of an issue.

After installing OverOps, it took about 15 minutes, and we found the error and saw what was causing it. OverOps helped turn days of work into minutes, quickly identifying and fixing different errors.

"OverOps gave us the exact variables and the user account in which the error had happened, allowing us to identify the problem and see that the user was not properly configured in the system"

Full code and variable state to immediately reproduce any error.
No need to manually reproduce issues by searching for information in logs.
Reduce MTTI by 90%+

<1% overhead in production
OverOps operates between the JVM and processor level enabling it to run in staging and production.

Proactive detection of all new and Critical errors
New issues are detected and routed to the right developer vs. discovered by users. Each error receives a unique code fingerprint unique it across the app.

No change to code or build
New issues are detected and routed to the right developer vs. discovered by users. Each error receives a unique code fingerprint unique it across the app.

Tel: 1-800-324-3053 www.overops.com
797 Bryant Street, San Francisco, CA 94107

Supported Platforms:

JDK 1.6 and above | HotSpot, OpenJDK, IBM JVM | Java, Scala, Clojure, Groovy | Linux, OS X, Windows | Docker, Chef, Puppet, Ansible | Coming soon: .NET

Integrations:

SLF4J, Log4j, Logback, Apache Commons Logging, Java Logger | Splunk, ELK, SumoLogic, and any other log management tool | AppDynamics, New Relic, Dynatrace | Workflow automation: Slack, HipChat, JIRA, Pagerduty | Webhooks | StatsD

Learn how OverOps can help you automate your deployments -
Schedule a demo with an OverOps monitoring engineer

Tel: 1-800-324-3053 www.overops.com
797 Bryant Street, San Francisco, CA 94107