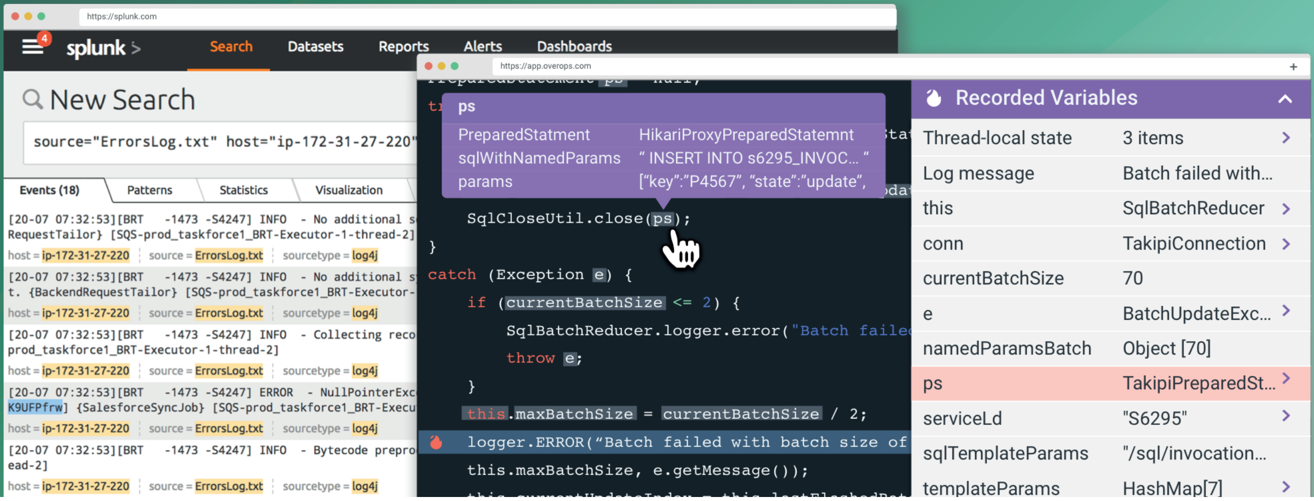
OverOps + splunk>

Log Files Suck. There’s a Better Way.

OverOps goes beyond log files to capture code-level insights for every error and slowdown in your applications and services – even the ones you didn’t see coming.



```
source="ErrorsLog.txt" host="ip-172-31-27-220"
```

```
[20-07 07:32:53][BRT   -1473 -S4247] INFO  - No additional s
RequestTailor} [SQS-prod_taskforce1_BRT-Executor-1-thread-2]
host = ip-172-31-27-220 | source = ErrorsLog.txt | sourcetype = log4j
[20-07 07:32:53][BRT   -1473 -S4247] INFO  - No additional s
t. {BackendRequestTailor} [SQS-prod_taskforce1_BRT-Executor-
host = ip-172-31-27-220 | source = ErrorsLog.txt | sourcetype = log4j
[20-07 07:32:53][BRT   -1473 -S4247] INFO  - Collecting reco
prod_taskforce1_BRT-Executor-1-thread-2]
host = ip-172-31-27-220 | source = ErrorsLog.txt | sourcetype = log4j
[20-07 07:32:53][BRT   -1473 -S4247] ERROR  - NullPointerExc
K9UFPfrw] {SalesforceSyncJob} [SQS-prod_taskforce1_BRT-Execu
host = ip-172-31-27-220 | source = ErrorsLog.txt | sourcetype = log4j
[20-07 07:32:53][BRT   -1473 -S4247] INFO  - Bytecode prepro
ead-2]
host = ip-172-31-27-220 | source = ErrorsLog.txt | sourcetype = log4j
```

ps	
PreparedStatment	HikariProxyPreparedStatemnt
sqlWithNamedParams	“ INSERT INTO s6295_INVOC... “
params	[“key”:”P4567”, “state”:”update”,

```
    SqlCloseUtil.close(ps);
}
catch (Exception e) {
    if (currentBatchSize <= 2) {
        SqlBatchReducer.logger.error("Batch failed
        throw e;
    }
   this.maxBatchSize = currentBatchSize / 2;
    logger.ERROR(“Batch failed with batch size of
    this.maxBatchSize, e.getMessage());
```

Recorded Variables	
Thread-local state	3 items
Log message	Batch failed with...
this	SqlBatchReducer
conn	TakipiConnection
currentBatchSize	70
e	BatchUpdateExc...
namedParamsBatch	Object [70]
ps	TakipiPreparedSt...
serviceLd	"S6295"
sqlTemplateParams	"/sql/invocation...
templateParams	HashMap[7]

FOX intuit COMCAST tripadvisor fiserv. Infosys

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

Identify the most critical issues in every release in test and production. **Prevent** them from being deployed. **Resolve** using the complete state of the code and environment.

OverOps runs in the cloud or on-premises, natively supporting microservice and containerized environments. With robust integrations – including OverOps links directly in your Splunk instance – OverOps ensures software reliability across the entire SDLC.

Identify all critical code errors and slowdowns in each release

---

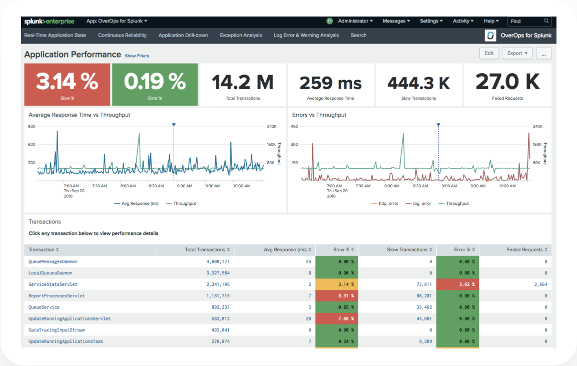
Prevent unreliable releases from being deployed into UAT and production

---

<> **Resolve** issues with complete code variables, DEBUG logs and host / container state at the moment of event

Visit **www.overops.com/splunk** to learn more.

Supercharge Your Splunk Instance

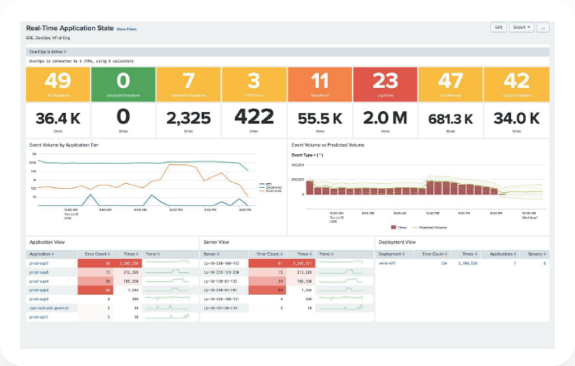


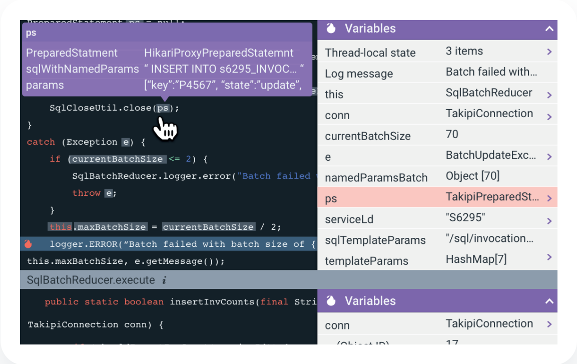
Identify Critical Issues – Even Ones Missed by Logs

OverOps automatically identifies critical code-level issues with the potential for impact, including new errors, increasing errors, slowdowns and more . We've created an app in Splunkbase so you can surface these insights directly in Splunk.

Prevent Poor Quality Releases

OverOps gathers complete information about every error and exception in your application and can map these to a release or a build. Through direct integration with your Splunk metrics dashboards, we expose data about the quality of your deployments so you can proactively prioritize and address critical issues – both in pre-production and production.





ps	
PreparedStatment	HikariProxyPreparedStatemnt
sqlWithNamedParams	" INSERT INTO s6295_INVOC… "
params	["key":"P4567", "state":"update",

```
    SqlCloseUtil.close(ps);
}
catch (Exception e) {
    if (currentBatchSize <= 2) {
        SqlBatchReducer.logger.error("Batch failed
        throw e;
    }
    this.maxBatchSize = currentBatchSize / 2;
    logger.ERROR("Batch failed with batch size of {
this.maxBatchSize, e.getMessage());
```

SqlBatchReducer.execute

```
   public static boolean insertInvCounts(final Stri
TakipiConnection conn) {
```

Variables	
Thread-local state	3 items
Log message	Batch failed with…
this	SqlBatchReducer
conn	TakipiConnection
currentBatchSize	70
e	BatchUpdateExc…
namedParamsBatch	Object [70]
ps	TakipiPreparedSt…
serviceLd	"S6295"
sqlTemplateParams	"/sql/invocation…
templateParams	HashMap[7]

Variables	
conn	TakipiConnection

Resolve Issues with Complete Context

OverOps links in Splunk allow you to immediately access deep context about the root cause of every error and exception so you can spend more time building and less time troubleshooting. Our data includes the complete variable state, last 250 Log statements including DEBUG and INFO level, event analytics and more.

Download the OverOps app from Splunkbase today: **splunkbase.splunk.com/app/4169**