

# *OverOps* & armory

- **Thank you for joining us!**  
**We will begin in just a few moments**

**OverOps** &  **armory**

● **To Go or Not To Go: 4 Error Metrics  
You Need for Confident Release  
Validation**

● November 19, 2020

# Welcome...



Lee Faus  
Field CTO



Eric Mizell  
VP, Solution Engineering



Alex Zhitnitsky  
Director, Product Marketing



# Thanks for joining!

**This is a LIVE webinar**



# Today's Agenda

## **The Speed vs. Quality Challenge**

- The growth of Continuous Delivery & how it's impacting software quality today

## **Understanding Canary Deployments & Decision Support**

- Intro to canary deployments and where they fall short in preventing production outages

## **Incorporating Error Metrics into your Canary Deployments**

- 4 code-level criteria you need to enhance your canary deployments

## **Hands-On Example**

- See what it looks like to add error metrics to Spinnaker canary analysis



# The Speed vs. Quality Challenge

**POLL:**

**How frequently do you  
release software?**

# Increased Velocity with Continuous Delivery



## Faster time to market

Releasing updates as they become ready means faster delivery of business value to customers, and automation means faster deployments.



## Higher quality

Multiple stages of automated verification reduce incidence of defects, and frequent customer feedback can become part of the product cycle.



## Lower risk

Frequent releases of small changes minimize the impact of any one change, and make root cause analysis easier, reducing recovery time.

## Software Quality Today

**20%**

Release new code at  
least once a day

**60%**

Of developer time  
wasted finding  
and fixing errors

**70%**

Prioritize quality over  
speed

**53%**

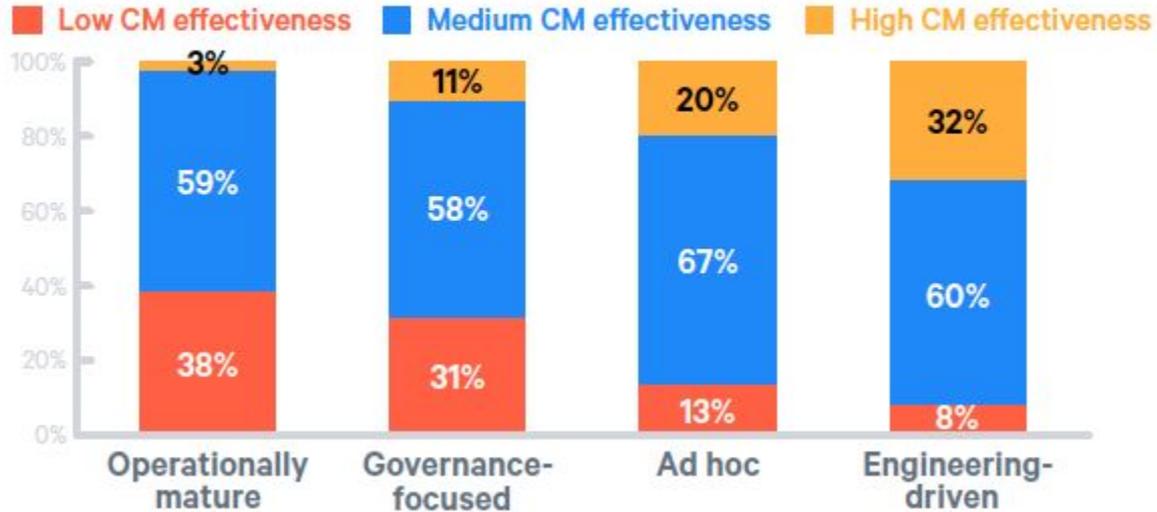
Experience incidents  
at least once a month



# **Understanding Canary Deployments & Decision Support**

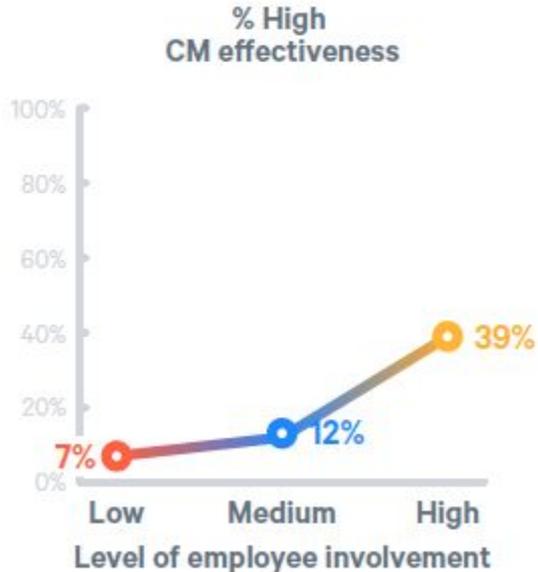
# DevOps Report 2020 Stats

## Change management approaches and level of effectiveness

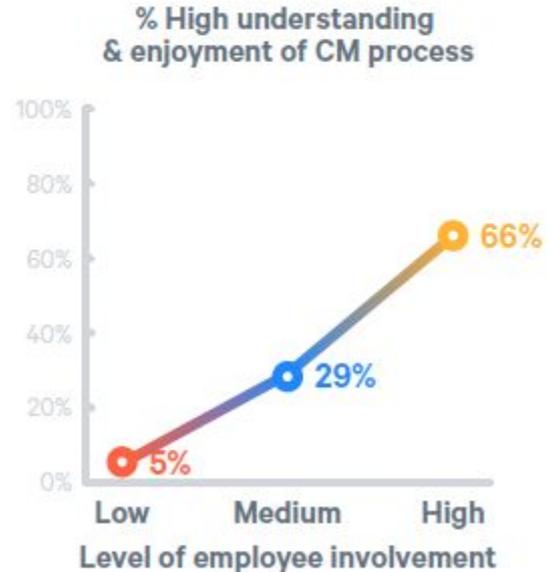


# DevOps Report 2020 Stats

Employee involvement and change management effectiveness



Employee involvement and understanding & enjoyment of CM process



# Understanding Decision Support



focus on resilience  
automate to dev environment  
operations driven

**slow**

safety



focus on consistency  
automate with guardrails  
data driven

**fast**

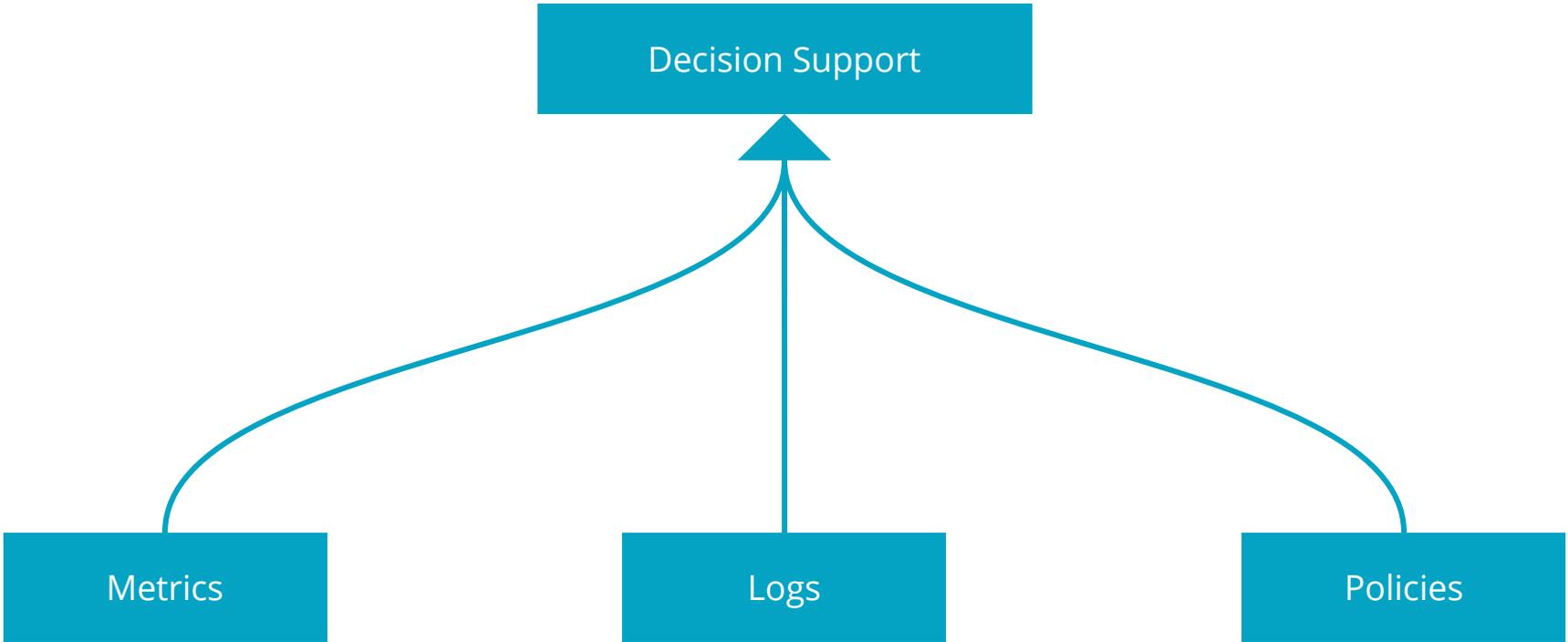
focus is status quo  
very little automation  
executive driven



risky

focus on velocity  
automate everything  
engineering driven





# The Canary Stage

## canary deploy (gitops)

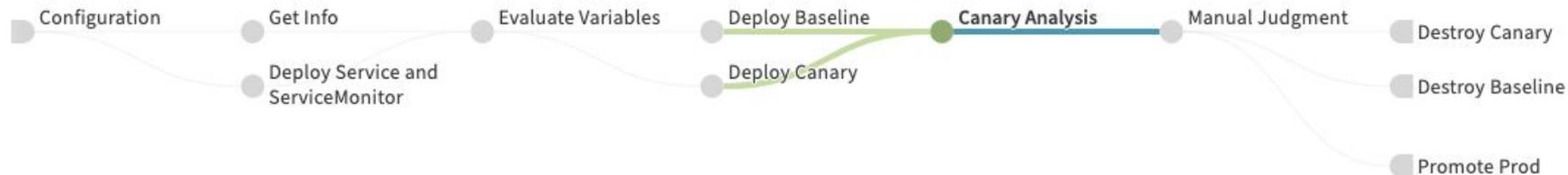
Permalink

Create

Configure

Pipeline Actions

This pipeline is locked and does not allow modification



Add stage

Copy an existing stage

# Configuration

## Canary Analysis Configuration

Analysis Config

**Analysis Type**   Real Time (Manual)  Retrospective

**Config Name**  

**Lifetime**   hours  minutes

**Delay**   minutes before starting analysis

**Interval**   minutes

**Step**  seconds

**Baseline Offset**   minutes

**Lookback Type**   

Baseline + Canary Pair 

**Baseline** 

**Baseline Location** 

**Canary** 

**Canary Location** 

# Execution of Canary Analysis

Execution Details

```
graph LR; A[Get Info] --> B[Evaluate Variables]; B --> C[Deploy Baseline]; C --> D[Canary Analysis (100)]; D --> E[Manual Judgment]; E --> F[Destroy Canary]; E --> G[Destroy Baseline]; E --> H[Promote Prod]; I[Deploy Service and ServiceMonitor] --> C;
```

**STAGE DETAILS: CANARY ANALYSIS**  
Duration: 05:16

Step	Started	Duration	Status
Canary Analysis	2020-10-16 06:03:16 PDT	05:16	SUCCEEDED

✓ CANARY ANALYSIS

Canary Summary	Canary Config	Task Status
		<b>Task</b>
		<b>Duration</b>
		✓ Interval Wait #1 01:00
		✓ Run Canary #1 00:02
		✓ Interval Wait #2 01:00
		✓ Run Canary #2 00:02
		✓ Interval Wait #3 01:00
		✓ Run Canary #3 00:02
		✓ Interval Wait #4 01:00
		✓ Run Canary #4 00:03
		✓ Interval Wait #5 01:00
		✓ Run Canary #5 00:02
		Aggregate Canary Results 00:00

# Canary Decision Criteria



## Infrastructure

CPU, memory, etc.



## Performance

Uptime, latency, etc.



## Errors

Total number of errors

# Today's Metrics Don't Paint the Full Picture



## **Infrastructure**

Lack insight into application level



## **Performance**

Reactive to customer impact



## **Errors**

Inability to classify which are new and critical



# **Incorporating Deeper Error Metrics into your Canary Deployments**

## Error Metric #1: New Errors

Did the release introduce any errors that didn't previously exist?



**Pass:**

No new errors are detected



**Fail:**

Release introduces a new error never previously seen

Detecting never before seen or new errors is hard and not always reliable.

Dynamic code analysis "fingerprints" your code and can detect new errors in live running applications.

As code moves through the CI/CD pipeline, dynamic code quality gates help identify all new errors that were introduced in the release.

## Error Metric #2: Critical Errors

Did the release introduce any severe or show stopping errors?



**Pass:**

No critical exception types detected



**Fail:**

The release introduces one or more predefined critical errors. In a Java application, some examples could be:

- NullPointerException
- IndexOutOfBoundsException
- InvalidCastException
- AssertionError

Critical errors can be defined by their exception type; NullPointerException, IndexOutOfBoundsException, DateParsing, YourCustomException, etc. Knowing if these kinds of errors exist in your CI/CD pipeline is important.

Many errors go undetected because nothing is written to the logs. When undetected, these silent "killers" can wreak havoc causing excessive debugging time.

## Error Metric #3: Resurfaced Errors

Did an error that was previously resolved appear again in the current release?



**Pass:**

No previously resolved errors are detected.



**Fail:**

Release resurfaces a previously fixed issue.

Once an error has been fixed, we don't expect it to resurface again. Unfortunately, this can happen from time to time.

Having the ability to detect resurfaced errors is critical. Knowing the error is resurfaced tells me code was merged in error and needs immediate attention prior to promotion.

## Error Metric #4: Unique Error Total

Did the release introduce an unusually high number of many unique (i.e. discrete) errors?



**Pass:**

Total number of unique errors falls below the standard baseline.



**Fail:**

Release introduces an unusually high number of unique errors.

Deduplicating errors can be a challenge as stack traces don't always tell the whole story.

With dynamic code analysis, you can de-duplicate errors at the bytecode level – meaning if new code is added above or below the offending line of code, the error is still deduplicated as same error and your unique count will not falsely increase.

# Hands-On Example

The background features a dark blue gradient with numerous thin, wavy lines in shades of teal and purple. These lines flow across the frame, creating a sense of motion and depth. The overall aesthetic is modern and digital.



## Canary

datadogmetrics

Edited: 2018-03-16 14:42:52 PDT

Add configuration

## CONFIGURATIONS

## METRICS

ALL CPU

METRIC NAME

CPU

Memory

Add Metric

## Configure Metric

Group	ERRORS
Name	New Errors
Fail on	<input checked="" type="radio"/> increase <input type="radio"/> decrease <input type="radio"/> either
Scope Name	default
OverOps Metric	total.errors.new
	<input type="button" value="Cancel"/> <input type="button" value="OK"/>

## SCORING

## Thresholds

Marginal

50

Pass

75

## Metric Group Weights

CPU

66

Memory

34

Canary

CONFIGURATIONS REPORTS

overopsmetric

66  
MARGINAL

**BASELINE**  
SCOPE  
availability-zone:us-west-2c,account:prod  
  
REGION  
notused

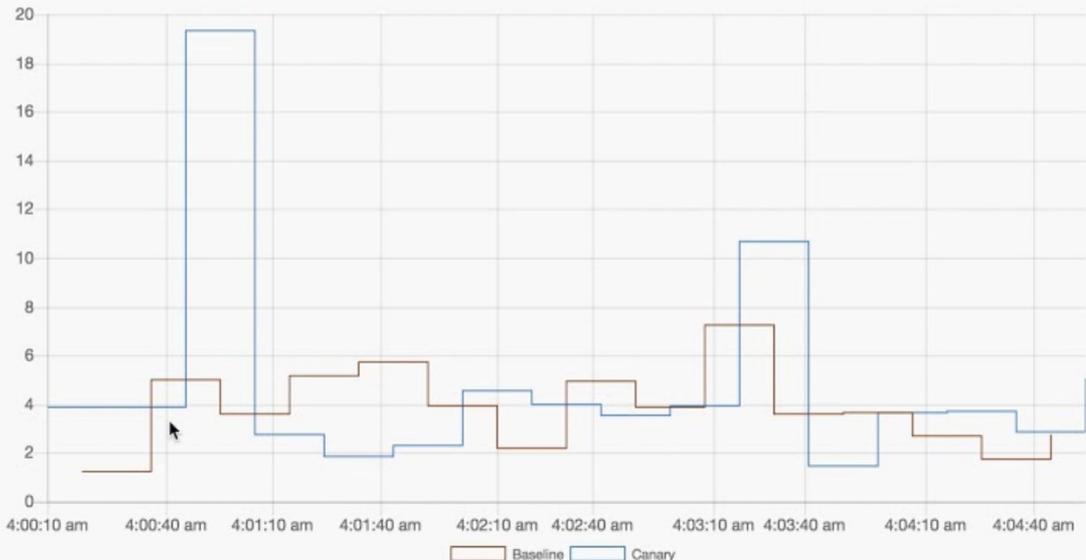
**CANARY**  
SCOPE  
availability-zone:us-west-2b,account:prod  
  
REGION  
notused

**TIME**  
START  
2018-03-08 04:00:00 PST  
END  
2018-03-08 04:05:00 PST  
  
STEP  
1 min

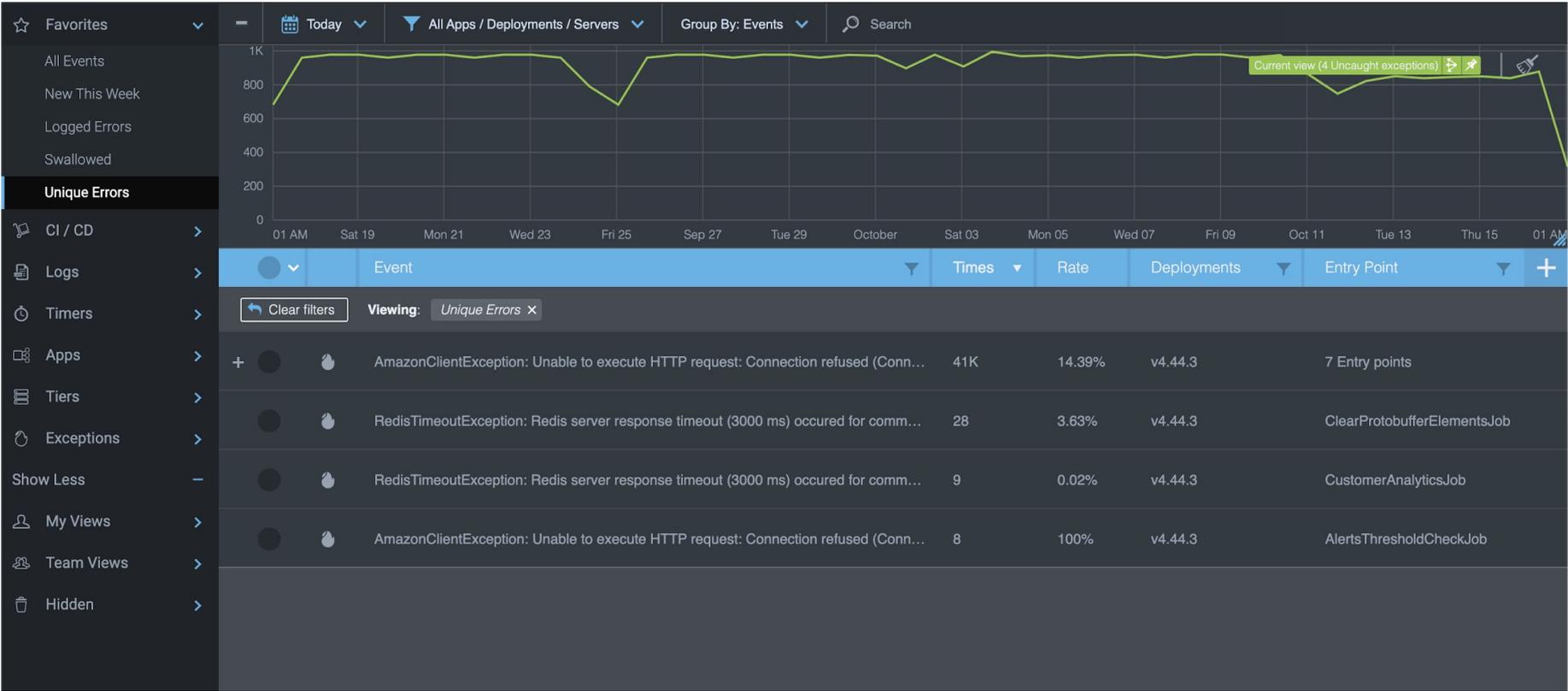
**THRESHOLD**  
MARGINAL  
50  
  
PASS  
75  
  
SOURCE  
Report  
Metrics

ALL CPU ERRORS MEMORY

METRIC NAME	RESULT
CPU	Pass
Memory	Fail
<input checked="" type="checkbox"/> Errors	Fail
<a href="#">Unique Error Total</a>	Fail
<a href="#">Critical Errors</a>	Fail
<a href="#">New Errors</a>	Pass
<a href="#">Resurfaced Errors</a>	Pass



NAME  
Unique Error Total





OverOps Prod

OverOps is active  
21 JVMs on 9 machines.

### Log Error Anomaly in BatchReduce

First seen: 14min ago



**CommonTablePartitioner.execute**

SqlPartitionOperation.internalExec...

SqlPartitionOperation\$1.internalEx...

SqlConnectionPool\$VoidTakipiCon...

SqlConnectionPool.internalWithCo...

SqlConnectionPool.WithConnection

SqlPartitionOperation.execute

SqlPartitionOperation.internalExec...

SqlPartitionOperation\$1.internalEx...



Kubernetes-dataStorage-AWS-East

All Servers / Apps / Deployments

Last day



</> Code Log JVM Resolve Label Add note

Logged Error: "Error creating S25558\_INVOCATIONS\_P\_2017\_06\_24 for S25558 INVOCATIONS"

### SqlBatchReducer.execute

```

72 Priya Tierney PreparedStatement ps = null;
73 try {
74 Elliot Matthams ps = RdbmsStatsInterface.get(
75 SqlRunUtil.executeBatch(ps, s
76 this.lastFlashedBatchIndex =
77 SqlCloseUtil.close(ps);
78 }
79 catch (Exception e) {
80 if (currentBatchSize <= 2) {
81 SqlBatchReducer.logger.er
82 throw e;
83 }
84 this.maxBatchSize = currentBa
85 Julia Rogers logger.ERROR("Batch failed wi
86 this.maxBatchSize, e.getMessage());
87 this.currentUpdateIndex = thi
88 SqlCloseUtil.close(ps);
89 }

```

**Julia Rogers**  
Commit date: April 1, 2020  
[Commit - Add new feature](#)

Recorded Variables	
Thread-local state	3 items
Log message	Batch failed with...
this	SqlBatchReducer
(Object ID)	16
BatchSize	70
CurrentUpdateIndex	70
lastFlashedBatchIn	0
maxBathIn	35
verboseLoggingEn...	false
conn	TakipiConnection
currentBatchSize	70
e	BatchUpdateException
namedParamsBatch	Object [70]

# Questions?



Lee Faus  
Field CTO



Eric Mizell  
VP, Solution Engineering



Alex Zhitnitsky  
Director, Product Marketing



**OverOps** &  **armory**

● **Thank you for joining us!**

● [overops.com/demo](https://overops.com/demo)

● [armory.io](https://armory.io)