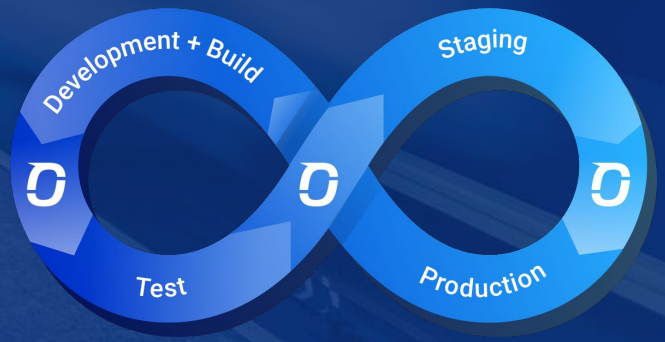
OverOps

Move Fast. Fix Faster.

& GitLab

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



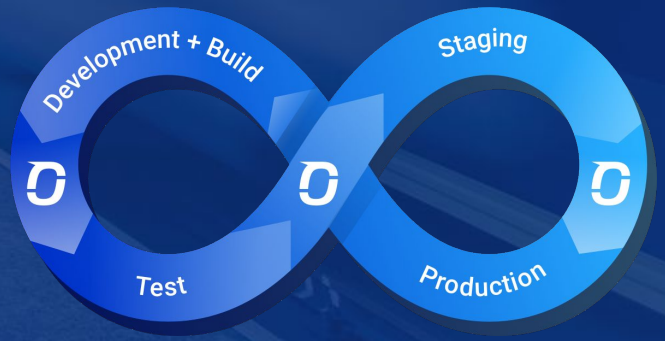
OverOps

Move Fast. Fix Faster.

& GitLab

Testing is Not Sufficient: 5 Ways to Know If Your Code is Really Production-Ready

July 22, 2020



Welcome...

Patrick Deuley

Sr. Product Manager, Ecosystem

GitLab

Eric Mizell

VP, Solution Engineering

OverOps

Alex Zhitnitsky

Director, Product Marketing

OverOps

Thanks for joining!

Copyright © 2020 OverOps. All rights reserved.

This is a live webinar! 😱
NOT pre-recorded
OverOps

Today's Agenda

- **The Need for a Unified CI/CD Toolchain**
- **Why Traditional Testing is Not Sufficient**
- **A Checklist for Production-Ready Code**
    - #1: Static Analysis
    - #2: New Errors
    - #3: Critical Exceptions
    - #4: Resurfaced Issues
    - #5: Unique & Total Error Volume
- **Enhance Your GitLab Pipeline with OverOps**

OverOps

Attendee Poll

What is your level of confidence when releasing new code to production?

Low, Medium, or High

OverOps

The Need for a Unified
CI/CD Toolchain
OverOps

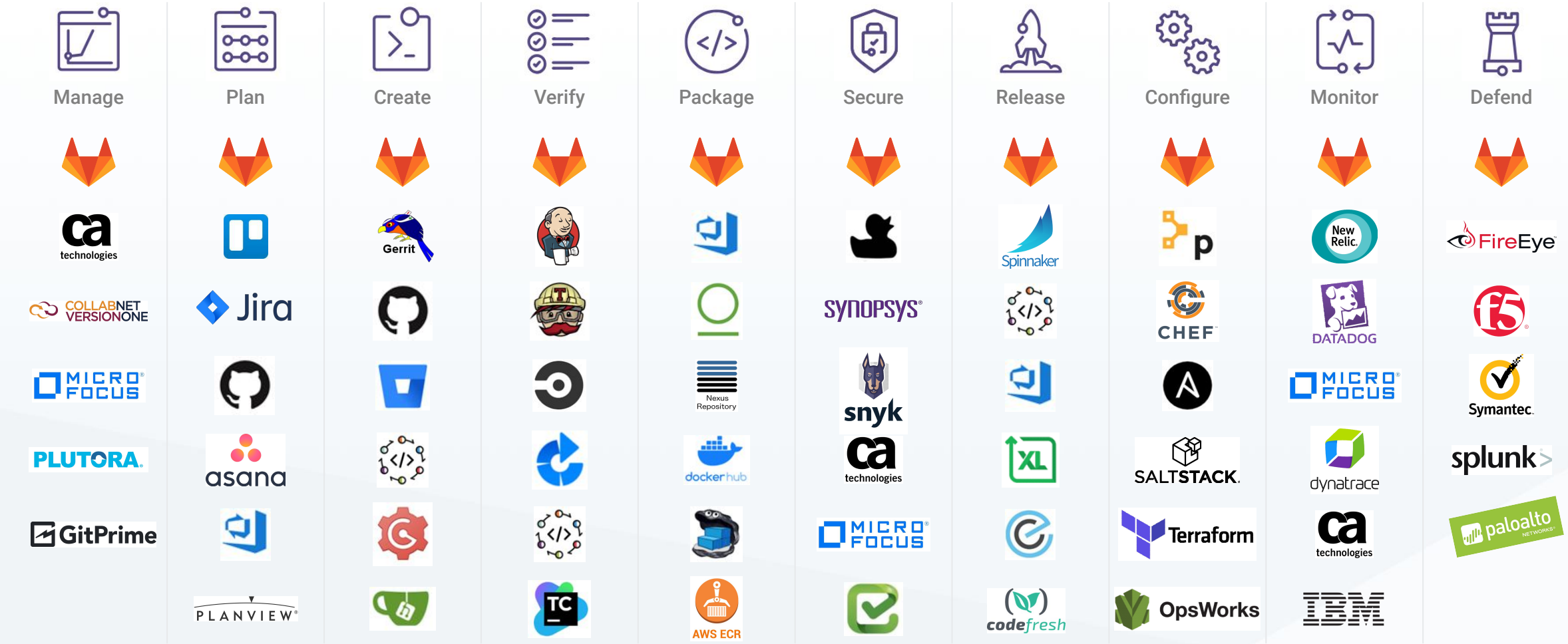
What is GitLab?

GitLab is a complete DevOps platform,

delivered as a single application that does everything from project planning and source code management to CI/CD, monitoring, and security.

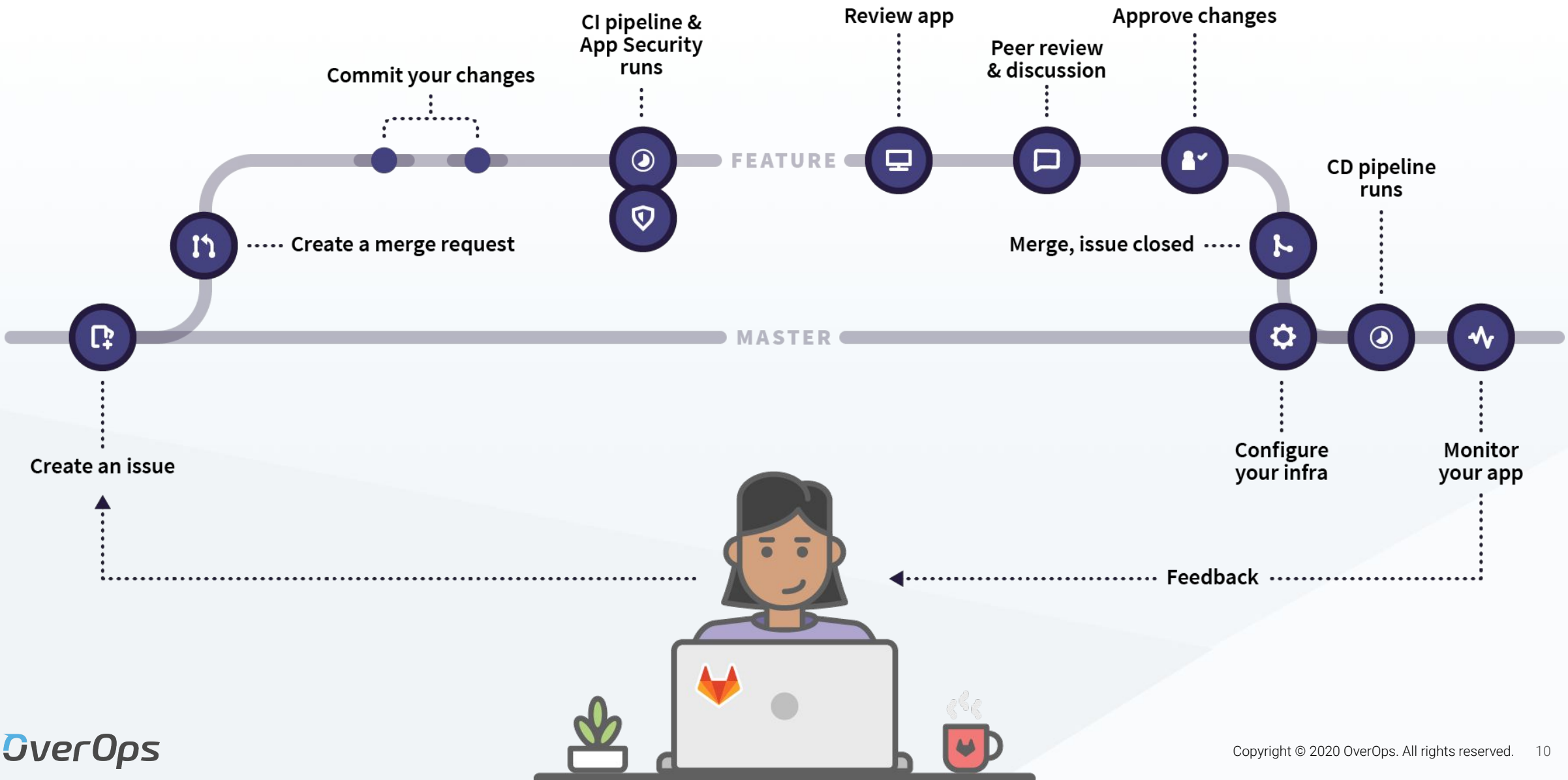
Copyright © 2020 OverOps. All rights reserved.

GitLab aims to replace multiple tools across the lifecycle



Copyright © 2020 OverOps. All rights reserved.

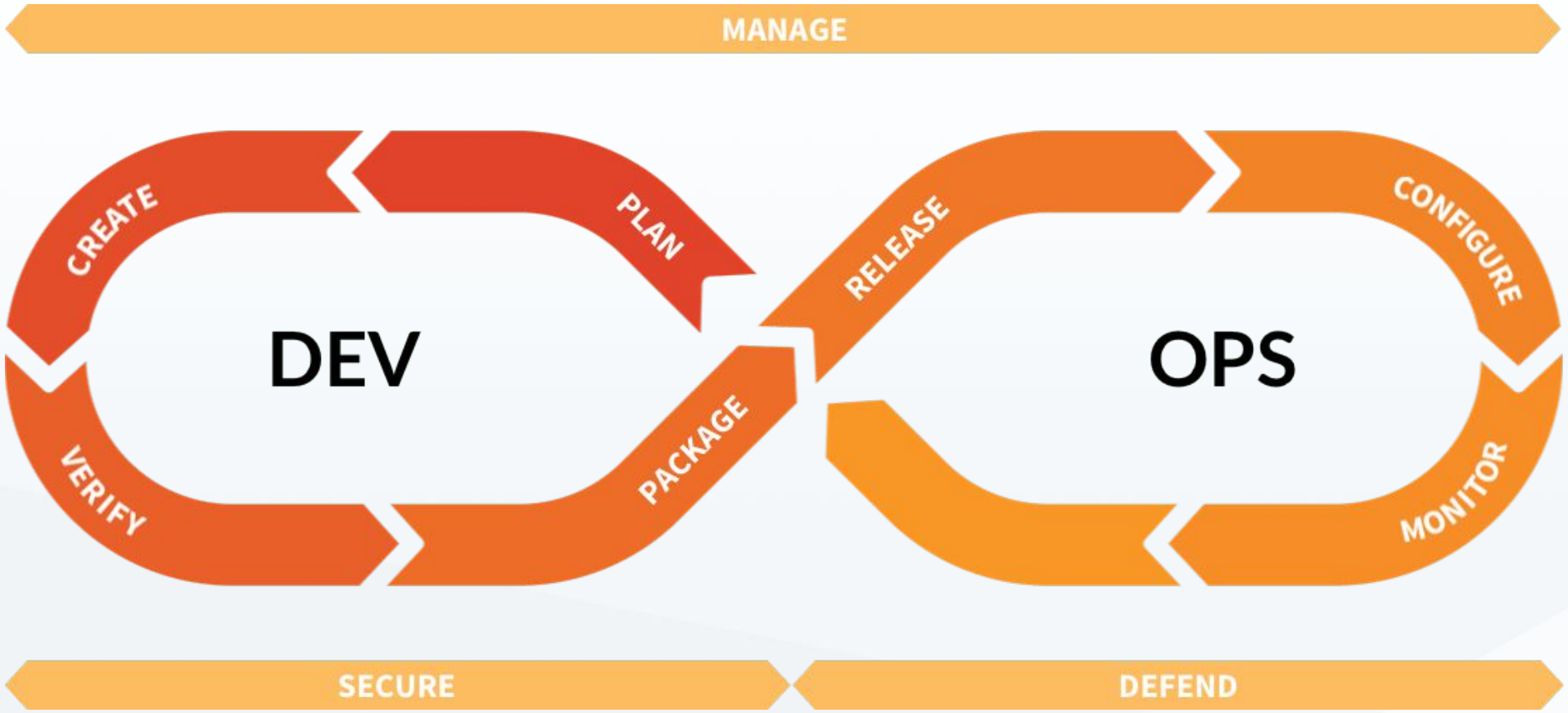
DevOps Best Practices Built-in



OverOps

Copyright © 2020 OverOps. All rights reserved.

The Complete DevOps lifecycle



Copyright © 2020 OverOps. All rights reserved.

GitLab “Dev” Section

Manage

- Access
- Compliance
- Import
- Analytics

Plan

- Project Management
- Portfolio Management
- Certify

Create

- Source Code
- Knowledge
- Static Site Editor
- Editor
- Gitaly
- Gitter
- Ecosystem (Me! 🎉)

Compliance Management

Upcoming: Adding features to enable an organization to enforce compliance controls inside of GitLab (e.g. Define MR approvals at a global level and only authorized users may change those settings)

https://about.gitlab.com/direction/manage/compliance-management/

Editor

Upcoming: Improvements to our newly-released VS Code plugin. Expanding our Web IDE support for more types of files, like specialized support for our `.gitlab-ci.yml` files.

https://about.gitlab.com/direction/create/web_ide/

OverOps

Copyright © 2020 OverOps. All rights reserved.

GitLab “Ops” Section

Verify
- Continuous Integration
- Runner
- Testing

Package

Release
- Progressive Delivery
- Release Management

Configure

Monitor
- APM
- Health

Continuous Integration

Our `rules` keyword allows for fine-grained and powerful logical controls over when a job runs, allowing you to evaluate against **109 predefined variables**, or any variables you provide directly.

https://docs.gitlab.com/ee/ci/yaml/index.html#rules

https://docs.gitlab.com/ee/ci/variables/predefined_variables.html

```
job:
  script: "echo Hello, Rules!"
  rules:
    - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'
      when: never
    - if: '$CI_PIPELINE_SOURCE == "schedule"'
      when: never
    - when: on_success
```

OverOps

Copyright © 2020 OverOps. All rights reserved.

GitLab “Enablement” Section

Enablement

- Distribution
- Geo
- Memory
- Global Search
- Database

Geo Replication

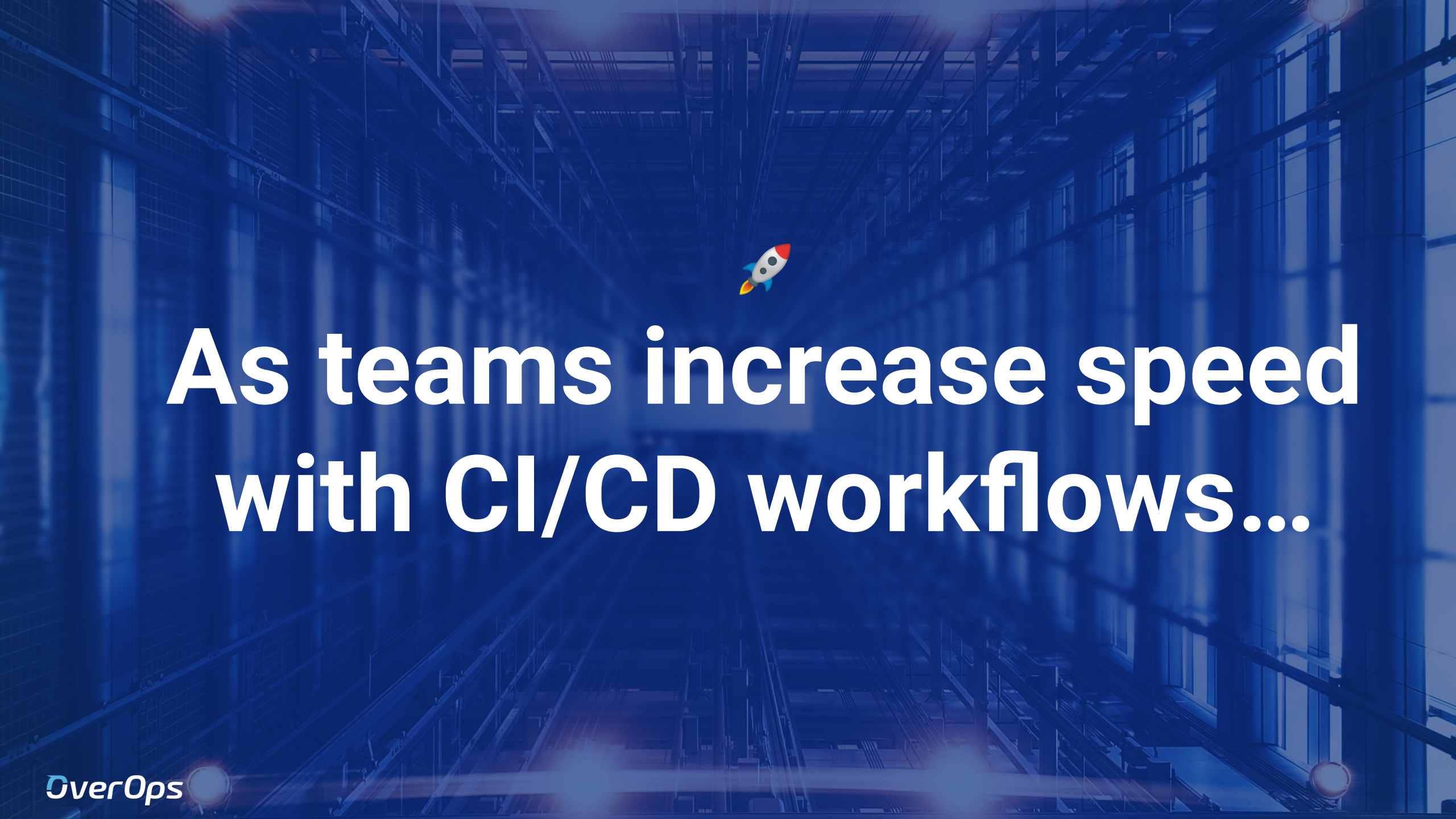
An incredibly powerful solution for widely distributed development teams. Adding read-only replication to secondary GitLab nodes can make clone and fetch tasks orders of magnitude faster, greatly improving the performance in that geographic area.

- Read-only replication to secondary nodes
- Reduces load times for automation, integrations, and workflows
- Can be used to quickly failover for disaster recovery (or planned failovers) to a secondary node

https://docs.gitlab.com/ee/administration/geo/replication/

Copyright © 2020 OverOps. All rights reserved.

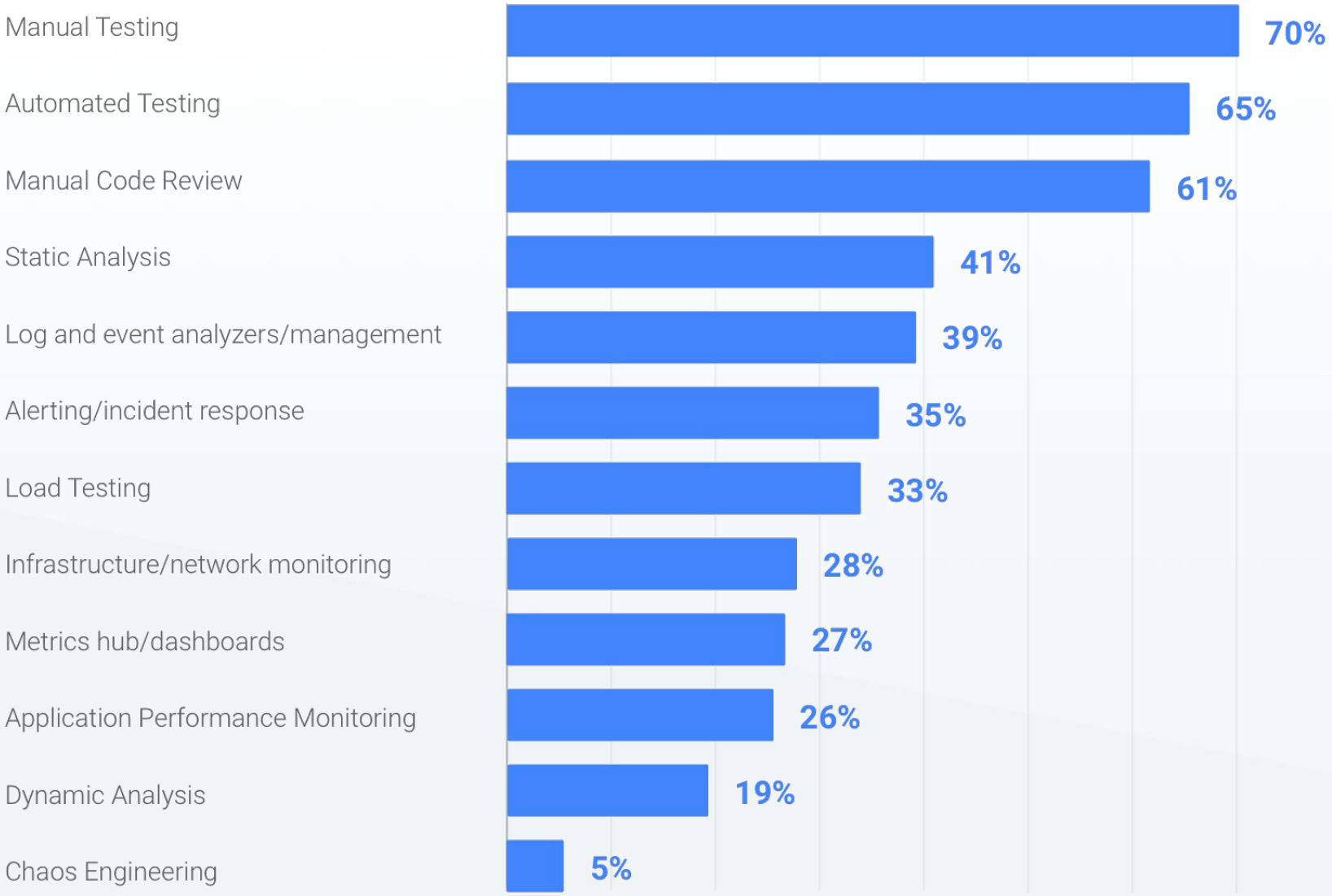
“Speed is important,
but quality is fundamental”
Pierre Bouchard
OverOps


🚀
As teams increase speed with CI/CD workflows...
OverOps

Quality becomes a greater challenge.

How are organizations addressing this today?

How are Organizations Addressing Quality in Pre-Production?



Manual testing is the

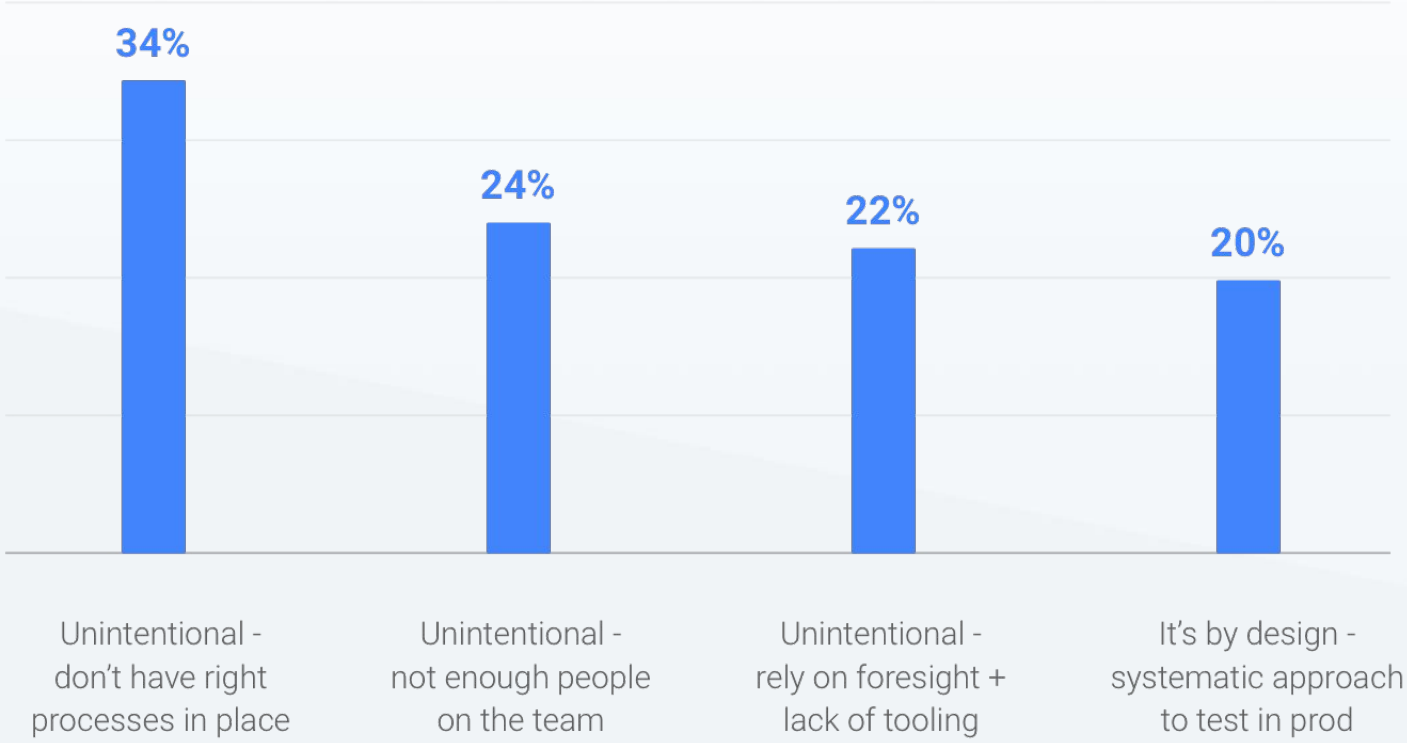
#1

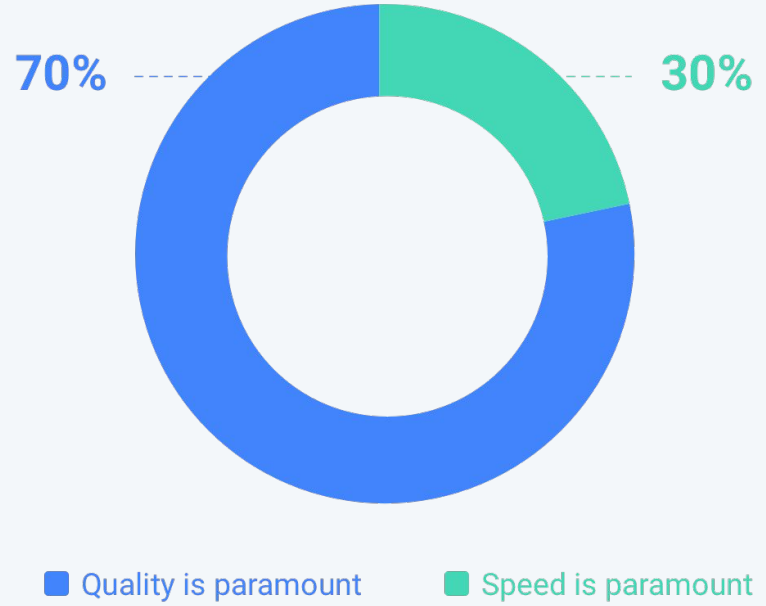
tool used in pre-production to ensure quality

Dynamic analysis is emerging as a new solution and monitoring tools are starting to slowly **shift left** into pre-production environments.

OverOps

What is the primary reason errors make it into production at your organization?





53%

of participants experience **critical/customer-impacting incidents at least once a month**

OverOps

🤔
What’s missing?

5 Code Quality Gates You Need in Your CI/CD Pipeline

OverOps

Code Quality Gate #1: Static Analysis

Can the code run? Does it follow best practices in readability/security?

- **Pass:**
  No style issues or code smells detected. No known security vulnerabilities

- **Fail:**
  Release introduces a new code smell or security vulnerability

Example Scenario:

In a recent change, you forgot to close a file descriptor which may cause memory leakage or you didn't wrap an if statement with parenthesis.

With static analysis, you can catch resource leaks and code style issues.

Copyright © 2020 OverOps. All rights reserved.

Code Quality Gate #2: 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

Example Scenario:

Your latest release in a mission-critical Java app introduces a 'ResourceNotFoundException' in a database that has never before experienced this issue.

Using the New Errors Quality Gate, your CI/CD tool automatically marks the release as unstable and blocks it from moving forward to production.

Copyright © 2020 OverOps. All rights reserved.

Code Quality Gate #3: Critical Exceptions

Did the release introduce any severe/showstopping 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

Example Scenario:

Your latest release for a Java application introduces a 'NullPointerException.'

Using the Critical Exceptions Quality Gate, your CI/CD tool automatically marks the release as unstable and blocks it from moving forward to production.

Copyright © 2020 OverOps. All rights reserved.

Code Quality Gate #4: 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.

Example Scenario:

You encounter a ParseException in a new release that you had previously addressed and resolved.

Using the Resurfaced Errors Quality Gate, your CI/CD tool automatically marks the release as unstable and blocks it from moving forward to production.

Copyright © 2020 OverOps. All rights reserved.

Code Quality Gate #5: Unique Error Volume

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.

Example Scenario:

Your latest release introduces 12 unique events. Your baseline value is set to 10 events.

Using the Unique Error Volume Quality Gate, your CI/CD tool automatically marks the release as unstable and blocks it from moving forward to production.

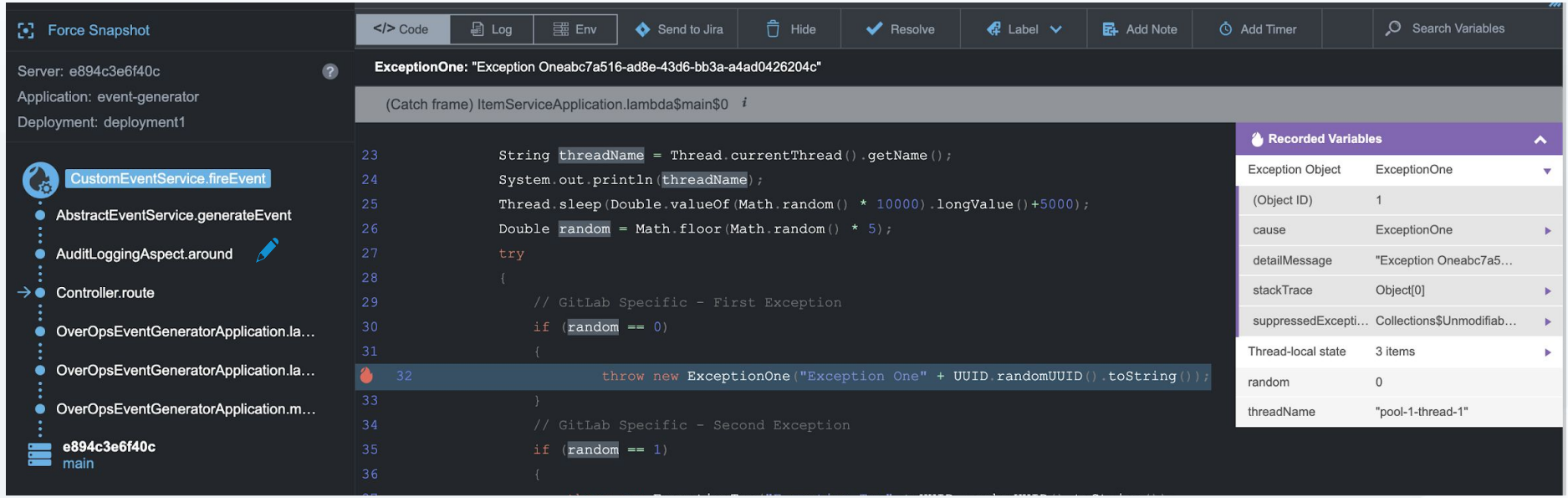
Copyright © 2020 OverOps. All rights reserved.

Enhance Your GitLab CI Pipeline with OverOps

Source Attach with GitLab

Get the complete context for every error

- See the original source code across the entire call stack.
- Correlate between commits and errors.
- Assign issues to the developer that introduced them.
- Link to Commit in source repository.

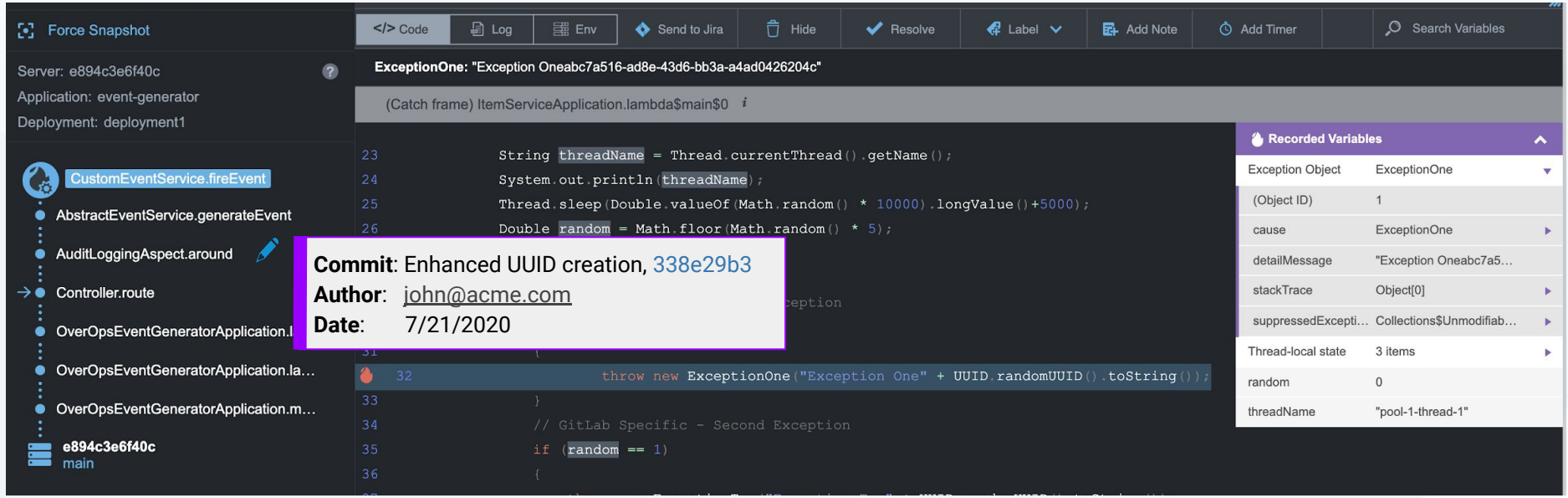


Copyright © 2020 OverOps. All rights reserved.

Source Attach with GitLab

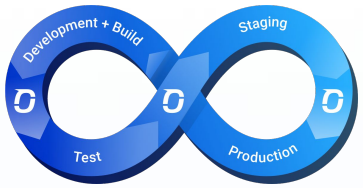
Get the complete context for every error

- See the original source code across the entire call stack.
- Correlate between commits and errors.
- Assign issues to the developer that introduced them.
- Link to Commit in source repository.



Copyright © 2020 OverOps. All rights reserved.

What's next?



Continuous Reliability for GitLab:

overops.com/technologies/gitlab/

Commit

Join us at GitLab Commit on August 26th, attend for free:

about.gitlab.com/events/commit/

Coming soon: Source Attach + Git Blame Support 🚀

OverOps

Questions?

Patrick Deuley

Sr. Product Manager, Ecosystem

GitLab

Eric Mizell

VP, Solution Engineering

OverOps

Alex Zhitnitsky

Director, Product Marketing

OverOps

overops.com/technologies/gitlab/

Copyright © 2020 OverOps. All rights reserved.

OverOps
&
GitLab
Thanks for attending!
overops.com
gitlab.com
overops.com/technologies/gitlab/
OverOps

2020 Report:

The State of Software Quality

Download the report:

bit.ly/overops-survey

OverOps