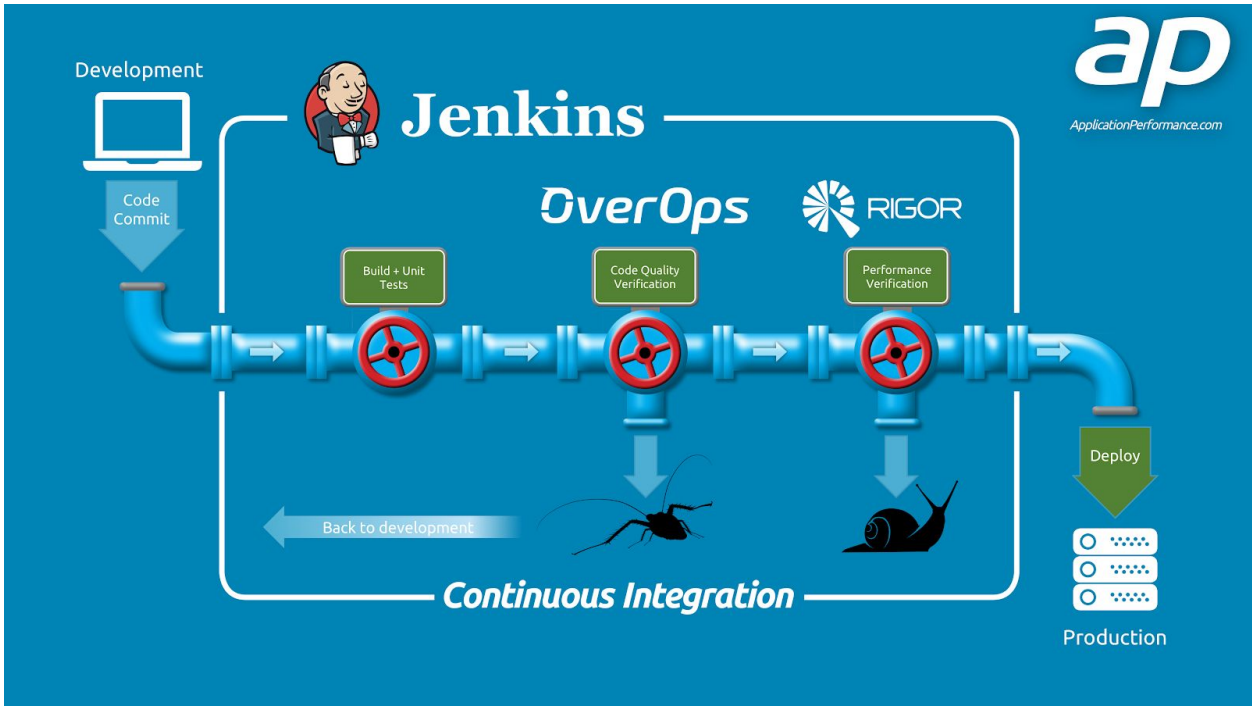
Stop bugs and slow code reaching production



Code Quality and Performance gates for your Continuous Integration Pipeline

Mick McGuinness
Technical Director
Application Performance Ltd.

May 2019

Contents

Introduction

We are living in a digitally connected world where consumers expect applications that respond quickly, without errors and deliver a great user experience.

And businesses need to innovate with the velocity of a startup or risk being disrupted by one. In short, businesses must build **faster**, **more reliable applications**, and **deliver them faster** than ever before in order to stay ahead of the competition.

However, the challenge of moving fast is far from simple. Businesses need to balance the speed and agility of innovation while at the same time maintaining quality, reliability and performance.

In this white paper, we will look at how Continuous Integration (CI) can help businesses to move faster and become more agile and we will discuss what we consider to be two essential features of your CI pipeline. That is:

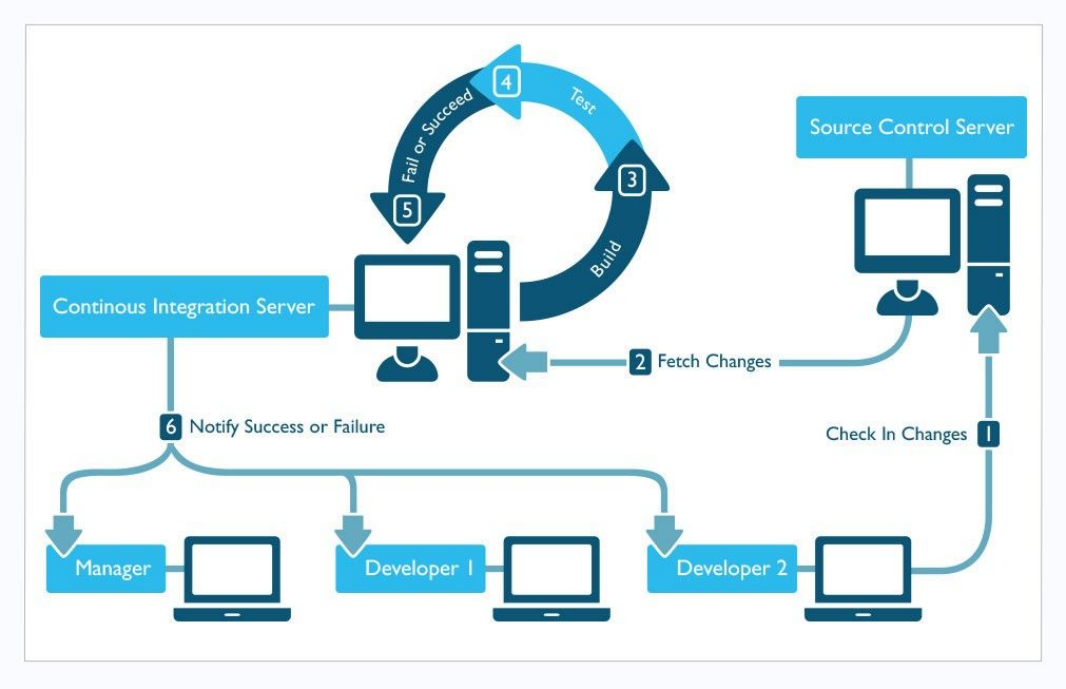
- **Continuous Reliability** - The practice of having code quality verification gates as part of your CI pipeline
- **Continuous Performance** - The practice of having performance verification gates as part of your CI pipeline

Continuous Integration

First, let's start by looking at Continuous Integration.

What is Continuous Integration?

Continuous integration (CI) systems provide automation of the software build and validation process driven in a continuous way by running a configured sequence of operations every time a software change is checked into the source code management repository. These are closely associated with agile development practices and closely related to the emerging DevOps toolsets.



1. Developer commits changes to the Source control server (E.g. git or subversion repository)
2. CI server fetches the changes
3. CI server automatically builds the new code (E.g. Ant, Maven, Gradle)
4. CI server performs a number of tests (E.g. Unit tests)
5. CI server marks the build as failed or succeeded
6. CI server notifies the developers and their manager about the status of the build

Step 4 above can include different kinds of tests or gates which need to succeed in order for the build to be marked as a success. This is where we can add something special.

Why use Continuous Integration?

In today's world, the ability to build quick, engaging applications is no longer a competitive advantage – it's an expectation. Now the advantage is to build fast, reliable experiences faster than everyone else. In this increasingly fast-paced industry, the need to push new features and experiences has outgrown traditional methods of software development, in which design, build, test, and implementation are discrete entities, which could be potentially spread out over a long timeline spanning weeks or months. Now with CI new features can be rapidly tested and deployed into production multiple times each day.

Benefits of CI

Some of the benefits of fast-paced release cycles:

1. Faster time to market for new innovations.
2. Reduced overhead on dev and ops through automated processes.
3. Helps collaboration between team members so recent code is always shared.
4. Releasing smaller incremental changes means earlier detection and prevention of defects.
5. Faster feedback on business decisions.

Potential drawbacks to CI

Moving fast with continual change can add risk if not done properly. These include:

1. Not enough time for proper testing.
2. More errors make their way to production.
3. More time trawling logs trying to debug errors.
4. More time trying to replicate problems in dev/test.
5. More time spent creating and deploying hotfixes.

What tools are available for Continuous Integration?

There are many options available for continuous integration. Here is a short list of some of the most popular:

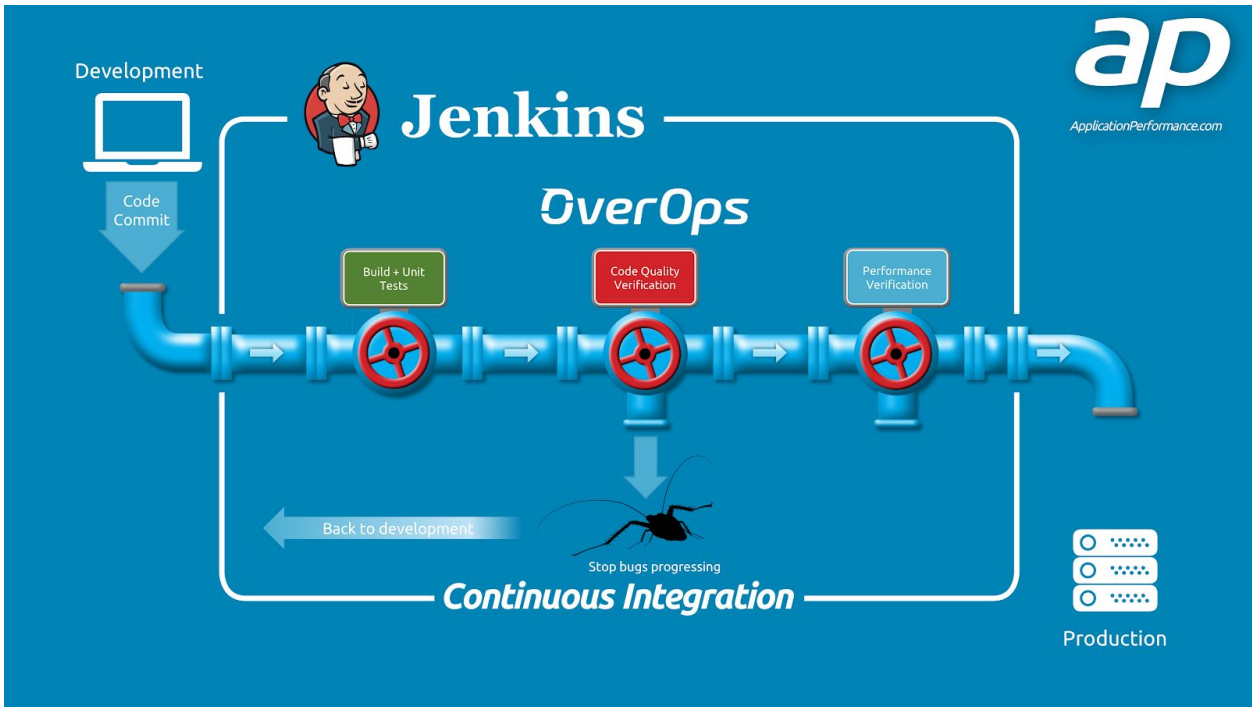
- **Jenkins** - Jenkins is an open-source CI tool written in Java. It is a cross-platform CI tool and it offers configuration both through GUI interface and console commands. What makes Jenkins very flexible is the feature extension through plugins.
- **TeamCity** - TeamCity is a mature CI server, coming from the labs of the JetBrains company, known for its tools like WebStorm and ReSharper. Despite being a Java-based solution, TeamCity offers the best .NET support among the tools on this list.
- **Bamboo** - Atlassian is a company focused on providing tools for software development teams and you might know them by their tools like JIRA and Bitbucket. Bamboo originally offered both cloud and On-premises solutions, but in the May 2016 the cloud version was discontinued in favour of the Bitbucket pipelines.
- **CircleCI** - Another cloud alternative that comes from the company with the same name. CircleCI currently only supports GitHub and the list of supported languages includes Java, Ruby/Rails, Python, Node.js, PHP, Haskell, and Scala.
- **Travis CI** - Travis CI is one of the oldest hosted solutions out there and it has won the trust of many people. Although it's mostly known for the hosted solution, it offers the on-premises version too.
- **GitLab CI** - GitLab CI is an integral part of the open-source Rails project GitLab from GitLab inc. It is hosted on GitLab.com, a free hosted service and it provides detailed git repository management with features like access control, issue tracking, code reviews and much more.

Continuous Reliability

Now, let's look at how you can continually test and verify **code quality** as part of your CI pipeline.

What is Continuous Reliability?

Continuous Reliability is the practice of continually testing the reliability and quality of your code as part of your CI pipeline. That means having an automated way to test the functionality of the application and check whether new errors or exceptions have been introduced by the latest build. The reliability can be tested in a number of ways including Unit tests, Integration tests and Functional tests to check the end-to-end application. These can be enhanced by static or dynamic code analysis tools such as OverOps.



The Continuous Reliability Process

- Trigger an automated test to detect new bugs, errors and exceptions
- Quality gate to fail the build if there are new bugs, errors or exceptions detected
- Stop bugs progressing to production

Why do we advocate Continuous Reliability?

Bugs that reach production are far more costly to fix. They can cause user frustration or worse still lead to lost revenue if key functionality is broken. The earlier you can detect a problem, the easier it is to resolve.

What tool do we recommend for Continuous Reliability?

AP recommends **OverOps**. OverOps is a platform which helps you identify and resolve critical errors with continuous code analysis and machine learning across your software delivery lifecycle. For any new or resurfaced errors, OverOps provides an Automatic Root Cause (ARC) link which takes you to the complete source code, variable and environment state behind any error, exception or timeout.

Benefits of OverOps

- **Automatic detection** of new issues and regressions.
- **Quality gate** to prevent bad code being promoted to production.
- **Intelligent routing** of feedback to the developer to speed up remediation.

An OverOps plugin is available for **Jenkins**.

Customers who trust OverOps

Learn more

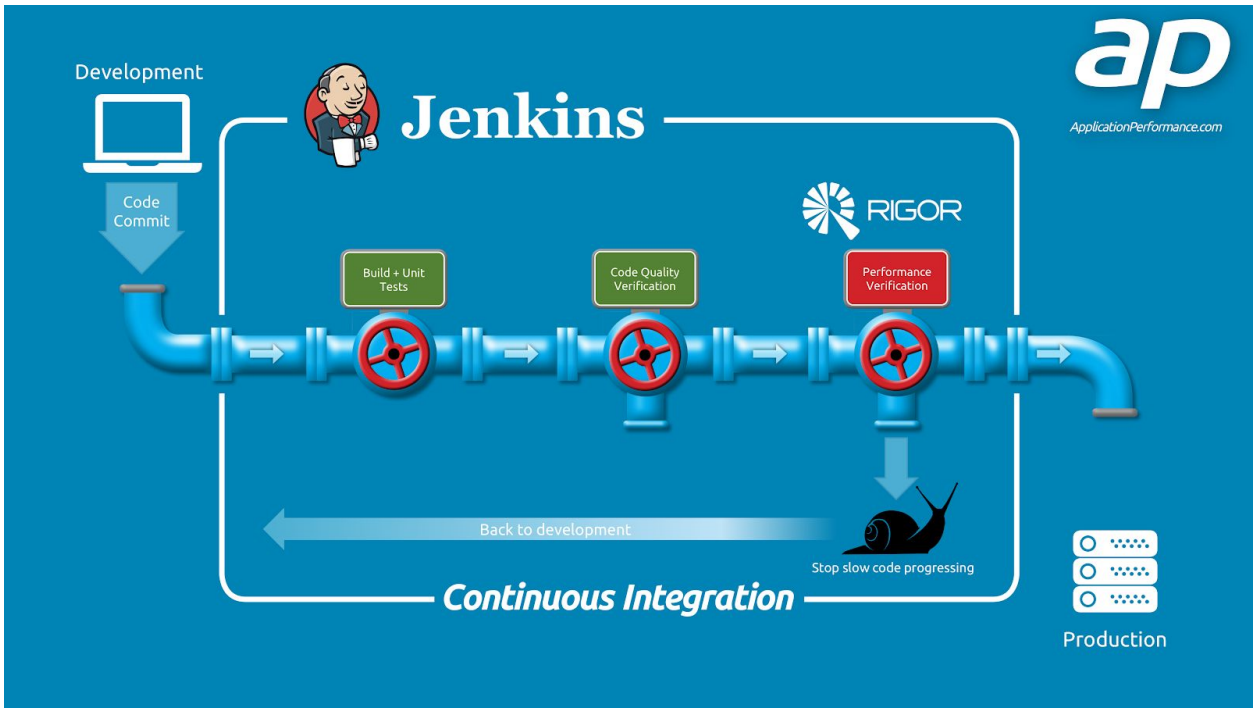
- To learn more about OverOps and Continuous Reliability please download the white paper titled "**How to Avoid the Hidden Costs of a CI/CD Workflow**" https://info.applicationperformance.com/overops-cicd-whitepaper
- Also, you might like to watch the **2-minute video** on our blog showing Jenkins and OverOps working together. https://blog.applicationperformance.com/stop-bugs-reaching-production-with-overops-and-jenkins
- Also, you can visit https://www.applicationperformance.com/overops/ to read more and **start a free trial**.

Continuous Performance

Next, let's look at how you can continually test and **verify performance** as part of your CI pipeline.

What is Continuous Performance?

Continuous performance is the practice of continually testing the performance of your code as part of your CI pipeline. That means having an automated way to test the performance of your web pages and APIs and measure how well optimised they are.



The Continuous Performance Process

- Trigger an automated performance test
- Performance gate to fail the build if performance doesn't meet the required levels.
- Stop slow code progressing to production

Why do we advocate Continuous Performance?

The worst time to learn about a business-critical performance issue is once it is released into production. Poor application performance can lead to losses in conversion rates and sends frustrated users running right to your competitors. The earlier you can detect a problem, the easier it is to resolve.

What tools do we recommend for Continuous Performance?

AP recommends **Rigor**. Rigor is a platform for digital experience monitoring and optimisation which equips you with actionable insights to optimise performance during your development cycle helping you to create and deliver faster.

Benefits of Rigor

- Validate performance against 280 best practice rules
- Performance gate to prevent slow code being promoted to production.
- Optimisation reports showing exact steps to improve performance

A Rigor plugin is available for **Jenkins**.

Customers who trust Rigor

Learn more

- To learn more about Rigor and Continuous Performance please download the white paper titled "**Building Faster Experiences with Continuous Performance**" https://www.applicationperformance.com/building-faster-experiences-with-continuous-performance-white-paper/
- Also, you can visit our website https://www.applicationperformance.com/rigor/ to read more and **start a free trial**.

Summary

Continuous Integration can help companies become more agile and release code faster than ever before. If you are already running a CI workflow then no doubt you will be running jobs for automated build and test. In addition, we recommend adding gates for code quality and performance? This is where OverOps and Rigor can help. They assess whether there have been any new bugs or performance regressions introduced by the current build and can fail the build if it determines those regressions are serious and provide insightful information to get you back on track, fast!

Both tools have plugins available for Jenkins. For those using other CI servers then creating code quality and performance gates can be achieved by calling into the APIs of the respective tools. Get in touch with us and we can assist you with this.

About AP

Calling on nearly 20 years of experience, we help Dev, Ops and Business users monitor and accelerate the performance of their critical applications, improve their code quality and deliver those applications faster through CI/CD.

About OverOps

OverOps analyses code in staging and production to automatically detect and deliver root cause for all errors – with no dependency on logging. It can be part of your CI/CD workflow to stop code defects reaching production.

About Rigor

Rigor not only continuously monitors your websites and API for availability, performance and errors but also assesses them against 280 best practices providing clear step-by-step instructions to optimise your site. It can be part of your CI/CD workflow to stop performance defects reaching production.