

Continuous Integration Tools

 BY MAX REHKOPP

Browse topics

Continuous Delivery Principles

Continuous Delivery Pipeline 101

What is Continuous Integration

Overview

How to get to Continuous Integration

How infrastructure as a service empowers the modern enterprise

3 Git Hooks for Continuous Integration

5 Tips for CI-Friendly Git Repos

Continuous Integration Tools

Trunk-based Development

Software testing for continuous delivery

What Is Continuous Deployment

Microservices and Microservice Architect

Bitbucket CI/CD tutorials

Continuous Delivery articles

Summary: Continuous integration (CI) is an agile and DevOps best practice that enables multiple developers to contribute and collaborate in a shared code base at a rapid pace. Without continuous integration, developer collaboration is a tedious manual process of coordinating code updates and merges.

Pillars of Continuous Integration

CI is built on agile and DevOps software development best practices including automated testing, version control, build automation, and automated deployments. Each one of these pillars of continuous integration has its own ecosystem of tools and philosophies. Software development tool providers have started offering full CI-as-a-service offerings which bundle these pillars into one holistic package. To better understand the bundled offerings lets review the foundation

Source Control Version Management

Automated Testing

Most serious software projects include an additional code base that is not explicitly responsible for the business product and features. This secondary code base is a test suite and acts as a set of assertions that assures the primary code base is working correctly without bugs.

During development, these tests are run by developers to validate that new code has not caused any regression on existing features. These test cases can also be run by extraneous tools to automate this validation process. CI service products will automatically run the test cases for a project on user-specified events. Generally, when a developer pushes code using the version control system an event will trigger the full test suite to run automatically.

Build Automation

“Builds” are the artifacts created to snapshot the current release version of a software project. Builds are distributed to end users through various networks. There is generally a set of scripted steps a project will take to create a build artifact. CI tools help streamline this build process with the use of automatic triggers from the version control system. One example trigger would be when new code is merged to the production branch of the codebase upload the build to a remote server for users to download.

Automated Deployments

When builds are ready to be distributed they go through a deployment process. Depending on the project deployment can have a variety of outcomes. For example, web projects are deployed to publicly accessible web servers. During this deployment, the artifact that was generated in the build phase is copied onto the web servers. The deployment process for mobile and desktop vary and may involve uploading to a ‘store’ so users may download the app.

Continuous Integration Tools Comparison

Bitbucket Pipelines

Summary: Bitbucket Pipelines is a CI tool directly integrated into Bitbucket, a cloud version control system offered by Atlassian. Bitbucket Pipelines is an easy next step to enable CI if your project is already on Bitbucket. Bitbucket Pipelines are managed as code so you can easily commit pipeline definitions and kick off builds. Bitbucket Pipelines, additionally offers CD. This means projects built with Bitbucket Pipelines can be deployed to production

RELATED TUTORIAL

Continuous Integration Tutorial

[Try this tutorial →](#)

SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)

infrastructure as well

Features:

- Easy setup and configuration
- Unified Bitbucket experience
- Cloud by 3rd party

Website: <https://bitbucket.org/product/features/pipelines>

Jenkins

Summary: Jenkins is a veteran CI Tool with a long proven track record. It is open source and driven by community updates. Jenkins is primarily intended to an on-premise installation. Jenkins is a great option when your organization needs on-prem support for handling sensitive customer like HIPAA compliance data.

Features:

- On-premise
- Open source
- Robust addon / plugin ecosystem

Website: <https://jenkins.io/>

AWS CodePipeline

Summary: Amazon Web Services is one of the most dominant cloud infrastructure providers in the market. They offer tools and services for all manner of infrastructure and code development tasks. CodePipeline is their CI Tool offering. CodePipeline can directly interface with other existing AWS tools to provide a seamless AWS experience.

Features:

- Fully cloud
- Integrated with Amazon Web services
- Custom plugin support
- Robust access control

Website: <https://aws.amazon.com/codepipeline/>

CircleCI

Summary: CircleCI is CI Tool that gracefully pairs with Github, one of the most popular version control system cloud hosting tools. CircleCI is one of the most flexible CI Tools in that it supports a matrix of version control systems, container systems, and delivery mechanisms. CircleCI can be hosted on-premise or used through a cloud offering.

Features:

- Notification triggers from CI events
- Performance optimized for quick builds
- Easy debugging through SSH and local builds
- Analytics to measure build performance

Website: <https://circleci.com/>

Azure Pipelines

Summary: Azure is Microsoft's cloud infrastructure platform, the Microsoft equivalent to Amazon Web Services. Like the aforementioned AWS CodePipeline, Azure offers a CI Tool that fully integrated into the Azure suite of hosting tools.

Features:

- Azure platform integration
- Windows platform support
- Container support
- Github integration

Website: <https://azure.microsoft.com>

GitLab

Summary: Gitlab is a new CI Tool which offers a full DevOps experience. Gitlab was created with intentions to improve Github's overall experience. Gitlab offers a modern UX with container support.

Features:

- On-prem or cloud hosting
- Continuous security testing
- Easy to learn UX

Website: <https://about.gitlab.com>

Atlassian Bamboo

Summary: Another CI offering from Atlassian. Whereas Bitbucket Pipelines is purely a cloud hosted option, Bamboo offers a self hosted alternative.

Features:

- Best integration with Atlassian product suite
- An extensive market place of add-ons and plugins
- Container support with Docker agents
- Trigger API for IFTTT functionality

Website: <https://www.atlassian.com/software/bamboo>

What is the Best CI Tool For Your Company?

When choosing a CI tool for your team there are a few important decision-making factors to consider.

Version Control System support

The core pillar of a CI / CD system is the support and integration of the underlying Version Control System (VCS). The most popular VCS's are Git, Subversion, Mercurial and Perforce. Cloud CI tools may offer support for some or all of these VCS's. It is critical to choose a CI tool that offers support for your projects VCS.

On-premises vs Cloud

Some of the aforementioned CI Tools like Jenkins can be installed on-premises. This means your team is responsible for configuring and managing the CI system on your own infrastructure. This can be beneficial for privacy and security reasons. For example, if you have customer data privacy concerns to meet compliance standards, on-premises may be a requirement. Additionally, on-premises instances may offer deeper customization and configuration options.

On-premises can be extra work that is not in line with your core business needs. Cloud options outsource the management of the CI Tool to a 3rd party vendor. The cloud hosting company then handles uptime, support, and scaling of the CI Tool allowing your team to focus on core business needs. This can be a huge benefit for tight budget teams or smaller companies that need aggressive focus on product market fit goals.

Container support

Containerization is a modern software development trend. Containerization enables the distribution of an immutable, repeatable, isolated copy of an application. Containerization is enabled by tools like Docker, and Kubernetes. Modern CI Tools will offer support for integrating containers into the CI/CD process. Containers solve the "works on my machine" problem. Ensuring that application code is packaged in a frozen snapshot of system level dependencies. This provides guarantees that when your team's code is executed on the CI Tool it is a replicate of the local environment. This eliminates a whole class of environment parity troubleshooting issues that arise without containers.

Plugins and 3rd party integrations

CI Tools become even more useful when integrated with the rest of your tech stack. Analytics about engineering team efficiency and performance can be collected from CI tools. Sprint planning applications can be tied to CI tools to automatically update sprint status when the code has been delivered. These integrations can be used to guide engineering team KPI's and roadmaps.

Get Started with CI

A modern CI tool is a critical requirement to enable a high performance engineering team. Engineering teams can optimize their communication and delivery speeds by using a CI tool. The market has many popular CI tool offerings. The pillar concepts to consider when choosing a CI tools are version control system support, on-premise vs cloud hosting, deployment pipelines, and external application integrations.

SHARE THIS ARTICLE



Microservices and
Microservices Architecture

Bitbucket CI/CD tutorials

Continuous Delivery articles



MAX REHKOPP

As a self-proclaimed "chaos muppet" I look to agile practices and lean principles to bring order to my everyday. It's a joy of mine to share these lessons with others through the many articles, talks, and videos I make for Atlassian

TUTORIAL

Continuous Integration Tutorial

This tutorial will show you how to get started with continuous integration in three simple steps.

[Try this tutorial →](#)

ARTICLE

Software testing for continuous delivery

Continuous delivery leverages a battery of software testing strategies to create a seamless pipeline that automatically delivers completed code tasks.

[Read this article →](#)

CI/CD Topics

Continuous Delivery
Continuous Integration
Continuous Deployment
Pipelines

Software Testing
Microservices
Tutorials

Sign up for more CI/CD articles and tutorials.

Email

[Subscribe](#)



Up Next
[Trunk-based Development →](#)