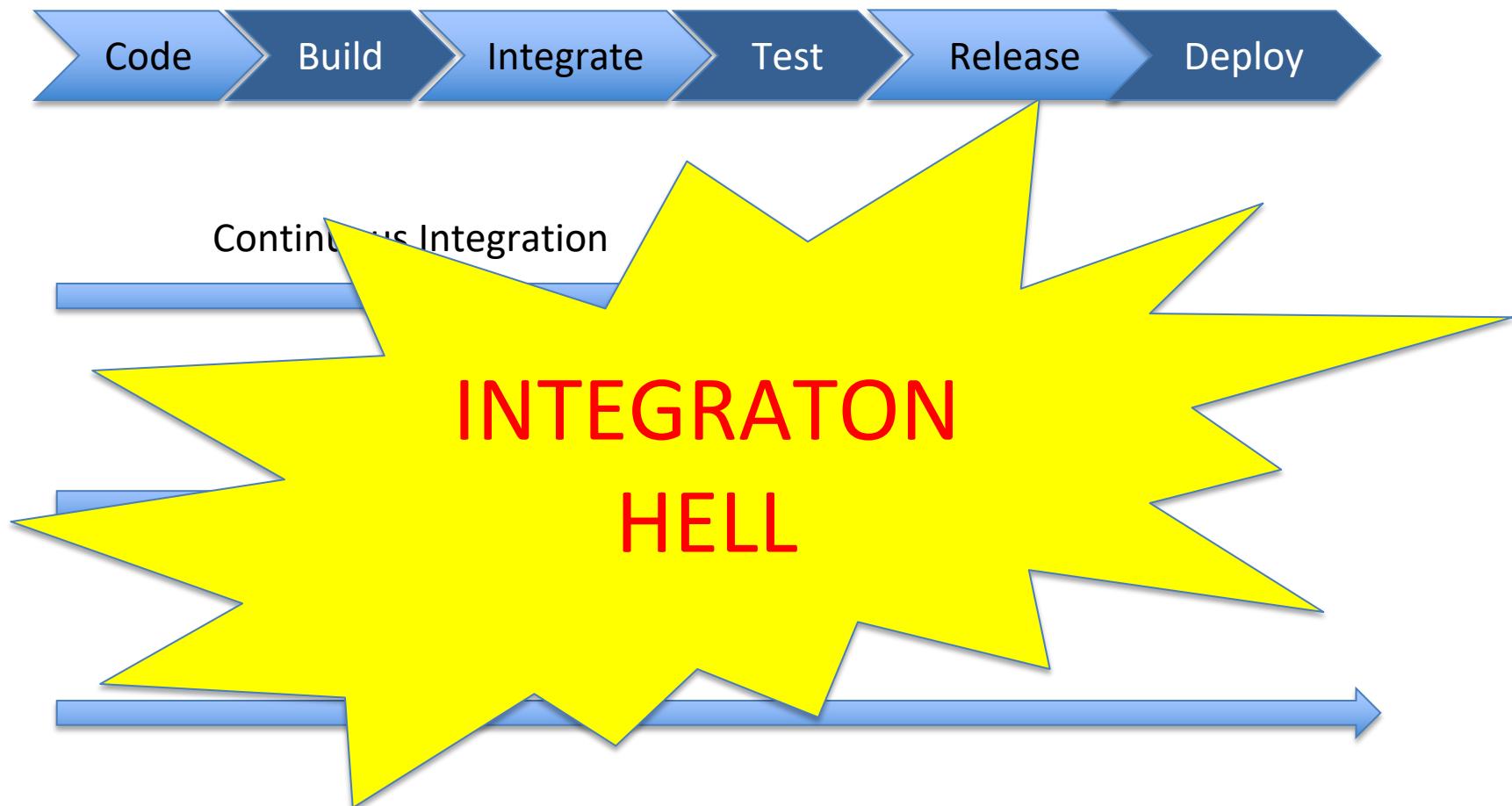


Continuous Integration with Jenkins

James Strong
Cloud Person and other stuff

Integrations



What is Continuous Integration Continuous Deployment?

Continuous – All day, Every day.

Integration – This is my code, that's your code, Let's get together.

Deployment – Deploy to ALL the servers.

Solution



Jenkins

Some Running Rules

1. Maintain a code repository
2. Automate the build
3. Keep the build fast
4. Make the build self-testing
5. Commit early, commit often
6. Every commit to the mainline gets built
7. Everyone can see the results of the build
8. Automate the deployment

What is Jenkins?

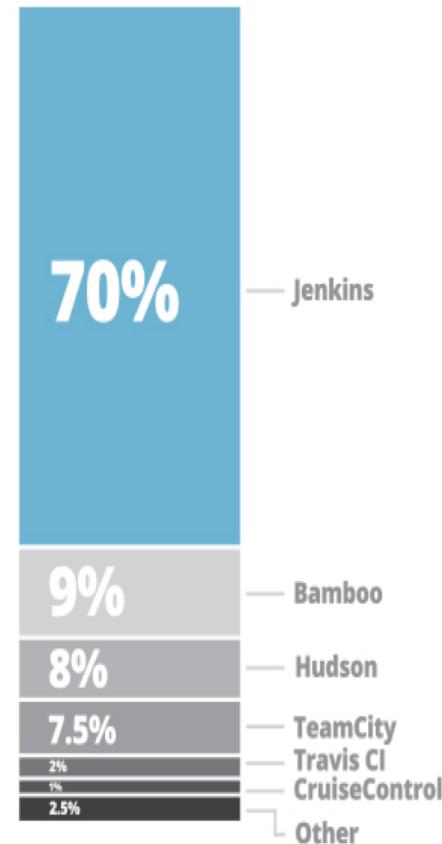
Some Facts:

- Written in Java.
- Jenkins was originally developed as the Hudson project. Hudson's creation started in summer of 2004 at Sun.
- Jenkins - , first released in Feb 2005
- CloudBees - Jenkins as a Service

Jenkins is just war file that schedules all things

- `java -jar jenkins.war`

Continuous Integration (CI) server used*

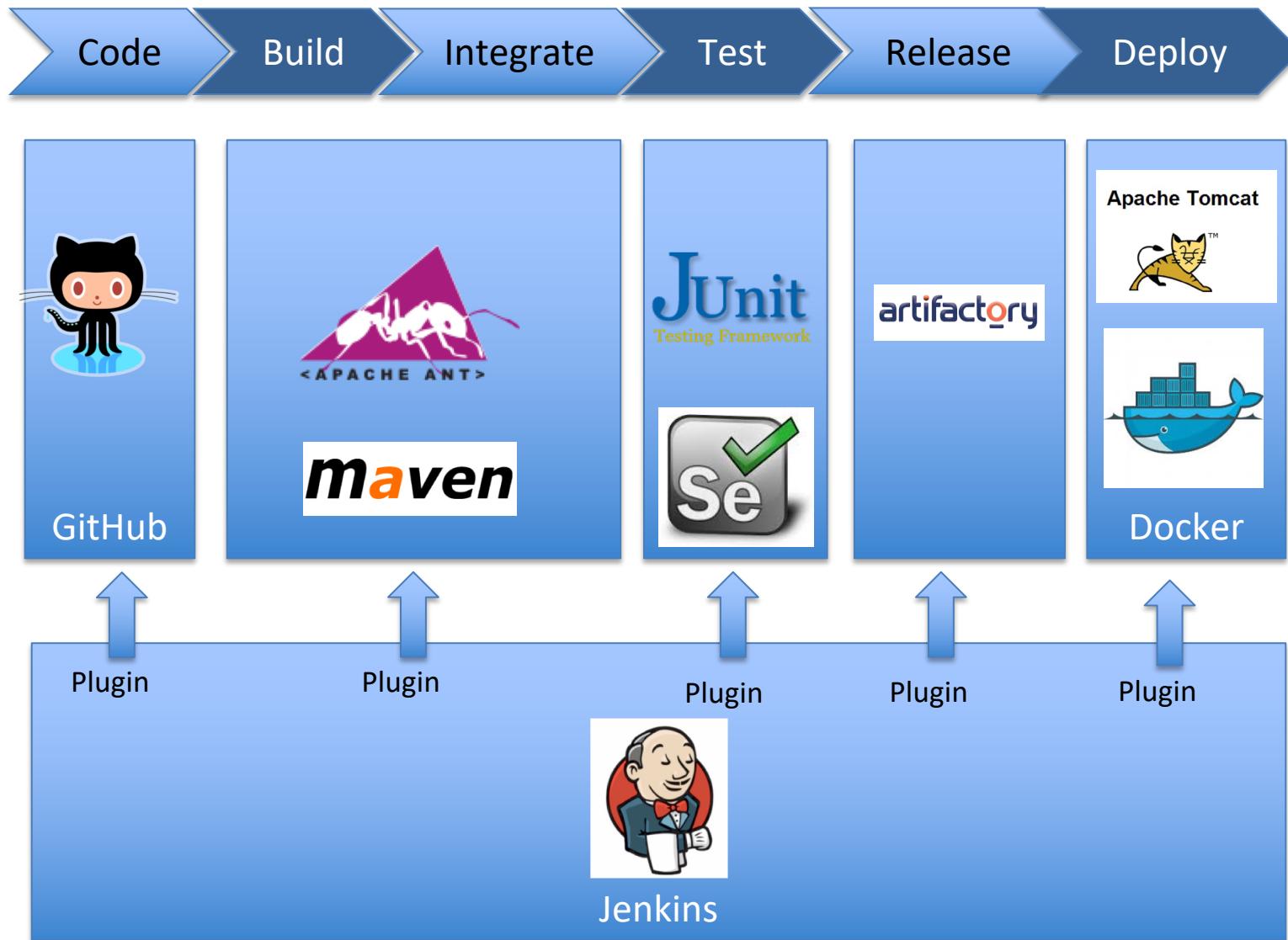


Why is Jenkins?

Why Jenkins?

- **Easy installation**
- **Easy configuration**
- **Rich plugin ecosystem**
- **Extensibility**
- **Distributed builds**

How Jenkins enables CI/CD



What is a Job/Item ?

- Basic currency of the Continuous Integration server
- Usually contains steps to compile, test, package, deploy your app and report
- Leverages plugins
- Establishes an Upstream-Downstream Relationship with other Jobs
- Can be scheduled or run manually
- Can run freestyle Bash commands. Anything you run in Linux you can run in Jenkins

The screenshot shows the Jenkins dashboard with the 'New Item' creation dialog open. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, Credentials, and My Views. The main area has a search bar and navigation for admin and log out. The 'New Item' dialog has an 'Item name' input field and several project type options:

- Freestyle project**: Described as the central feature of Jenkins, combining any SCM with any build system.
- Maven project**: Builds a Maven project using POM files.
- External Job**: Allows recording execution of processes outside Jenkins, even on remote machines.
- Multi-configuration project**: Suitable for projects with many configurations, like testing on multiple environments.
- Copy existing Item**: An option to copy an existing item from another source.

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Anatomy of a Job

Project Name/Description



Maven project name

Description

[Escaped HTML] [Preview](#)

Plugin



Git
Repositories

Repository URL

Build and Post Build Steps



Build

Root POM

Goals and options

`clean install -P dev -DskipTests=true`

Post Steps

Run only if build succeeds Run only if bu

Should the post-build steps run only for successful builds, etc.

Job chaining

Trigger/call builds on other projects

Build Triggers

Projects to build

Notification



E-mail Notification

Recipients

Plugins



Customize Jenkins

Choose from 1103 plugins to customize Jenkins exactly to your needs.

- Help Jenkins to interface with other tools to make CI possible
- Can be installed/updated using Jenkins web UI or by hand
- Developers can create their own plugins if needed (rarely if ever)

4 Plugins by topic

- 4.1 [Source code management](#)
- 4.2 [Build triggers](#)
- 4.3 [Build tools](#)
- 4.4 [Build wrappers](#)
- 4.5 [Build notifiers](#)
- 4.6 [Slave launchers and controllers](#)
- 4.7 [Build reports](#)
- 4.8 [Artifact uploaders](#)
- 4.9 [Other post-build actions](#)
- 4.10 [External site/tool integrations](#)
- 4.11 [UI plugins](#)
- 4.12 [List View column plugins](#)
- 4.13 [Page decorators](#)
- 4.14 [Authentication and user management](#)
- 4.15 [Cluster management and distributed build](#)
- 4.16 [CLI extensions](#)
- 4.17 [Maven](#)
- 4.18 [Parameters](#)
- 4.19 [iOS development](#)
- 4.20 [.NET development](#)
- 4.21 [Android development](#)
- 4.22 [Ruby development](#)
- 4.23 [Library plugins](#)
- 4.24 [Scala plugins](#)
- 4.25 [Misc](#)
- 4.26 [Uncategorized plugins](#)

<https://wiki.jenkins-ci.org/display/JENKINS/Plugins>

Github Webhooks

- Webhooks provide a way for notifications to be delivered to an external web server whenever certain actions occur on a repository or organization.
 - A repository is pushed to
 - A pull request is opened
 - A GitHub Pages site is built
 - A new member is added to a team
- Trigger a build when a change is pushed to GitHub
- This feature enables builds after post-receive hooks in your GitHub repositories
 - Manual
 - Automatic

<https://wiki.jenkins-ci.org/display/JENKINS/GitHub+Plugin>

<https://help.github.com/articles/about-webhooks/>

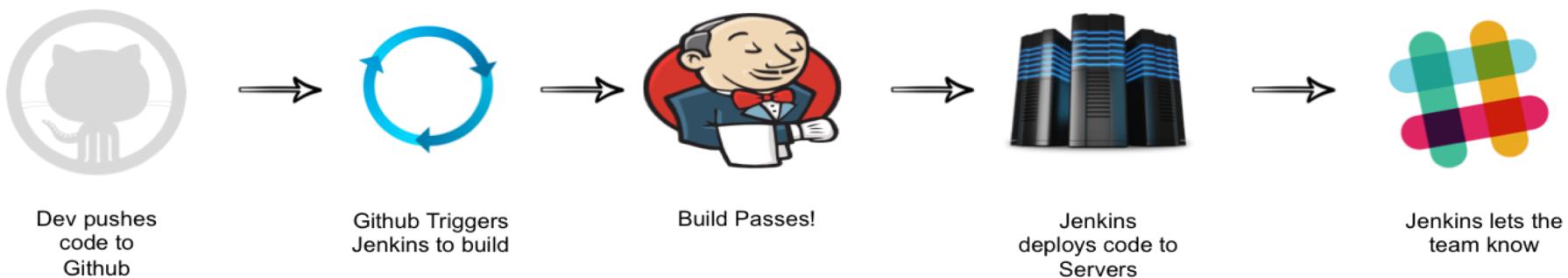
Job Notifications

- Slack
- E-mail
- PagerDuty

pagerduty



Bringing them together



THE END

**PUT ALL THE
THINGS IN**

JENKINS

More Information

<https://jenkins-ci.org/>

<https://dzone.com/refcardz/jenkins-paas>

<https://www.cloudbees.com/jenkins/about>

<https://yow.eventer.com/events/1004/talks/1062>