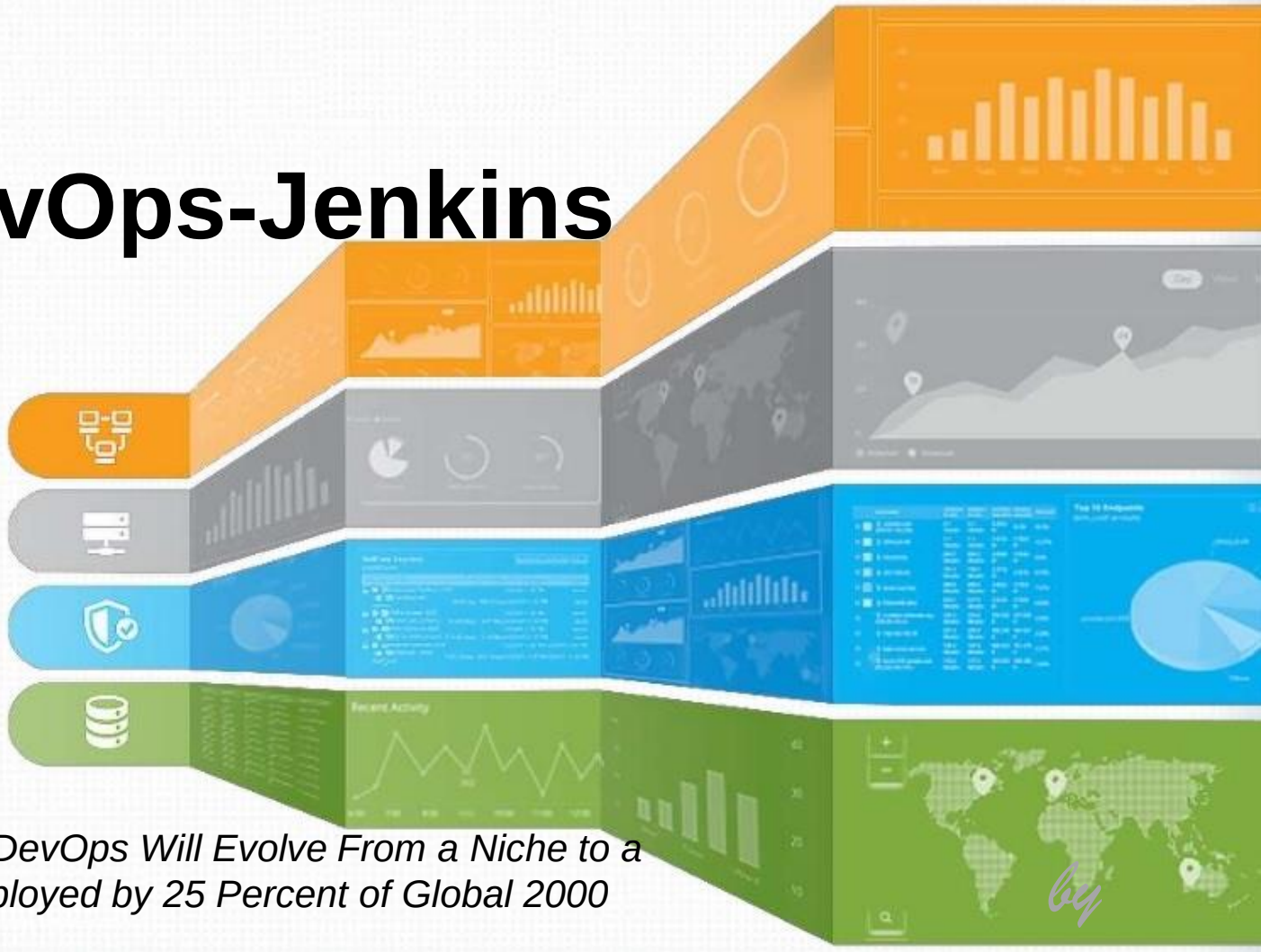
DevOps-Jenkins



As per Gartner By 2019, DevOps Will Evolve From a Niche to a Mainstream Strategy Employed by 25 Percent of Global 2000 Organizations.

Jenkins

“Continuous Build System”

Agenda

- Continuous Integration (CI)
  - What is it?
  - What are the benefits?
  - Continuous Build Systems
- Jenkins
  - What is it?
  - Where does it fit in?
  - Why should I use it?
  - What can it do?
  - How does it work?
  - Where is it used?
  - How can I get started?
- Putting it all together
- Conclusion
- References

CI - Defined

“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible” – Martin Fowler

CI- What does it really mean?

At a regular frequency (ideally at every commit), the system is:

- **Integrated**

  All changes up until that point are combined into the project

- **Built**

  The code is compiled into an executable or package

- **Tested**

  Automated test suites are run

- **Archived**

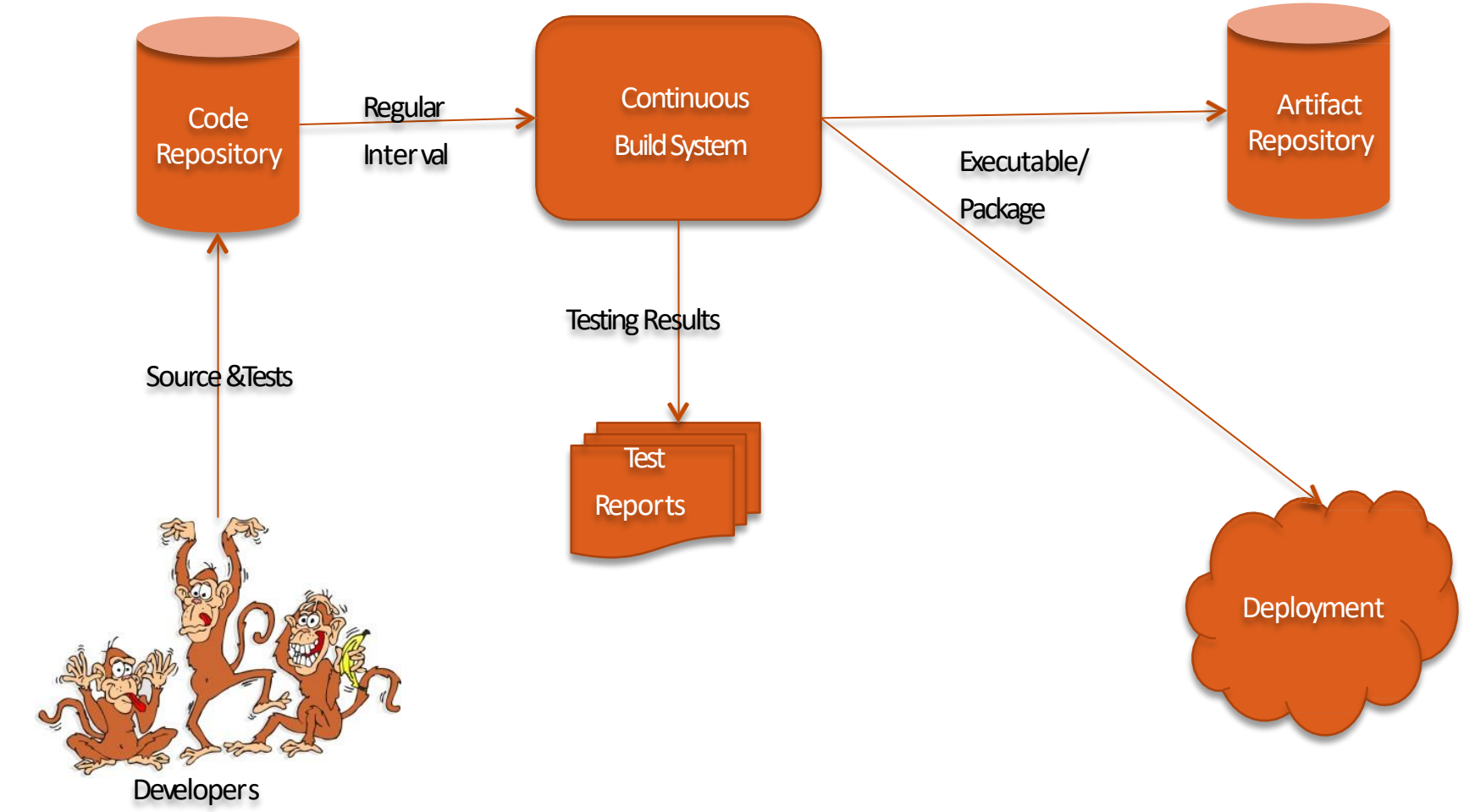
  Versioned and stored so it can be distributed as is, if desired

- **Deployed**

  Loaded onto a system where the developers can interact with it

JENKINS

CI - Workflow



CI – Benefits

- Immediate bug detection
- No integration step in the lifecycle
- A deployable system at any given point
- Record of evolution of the project

CI – The tools

- **Code Repositories**

  SVN, Mercurial, Git

- **Continuous Build Systems**

  Jenkins, Bamboo, Cruise Control

- **Test Frameworks**

  JUnit,Cucumber, CppUnit

- **Artifact Repositories**

  Nexus, Artifactory, Archiva

Jenkin

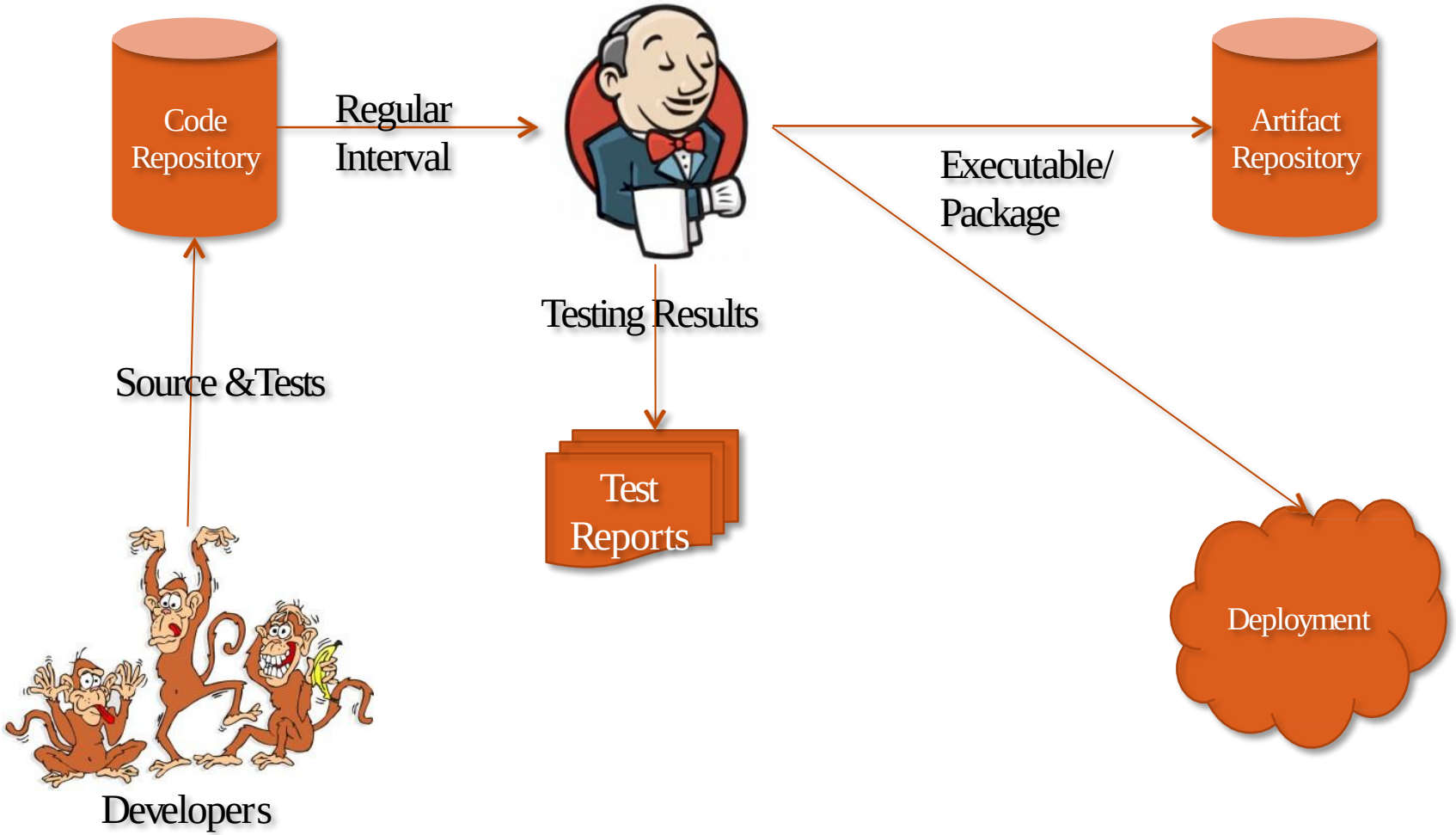
- Branched from Hudson
- Java based Continuous
- Build System Runs in servlet container
  - Glassfish, Tomcat
- Supported by over 1500+ plugins
  - SCM, Testing, Notifications, Reporting, Artifact Saving, Triggers, External Integration
- Under development since 2005
- https://jenkins.io/

Jenkins - History

- 2005 - Hudson was first release by Kohsuke Kawaguchi of Sun Microsystems
- 2010 – Oracle bought Sun Microsystems
- Due to a naming dispute, Hudson was renamed to Jenkins
- Oracle continued development of Hudson (as a branch of the original)

JENKINS

Jenkins – Fitting in



Why Jenkins? Flexibility!

- Jenkins is a highly configurable system by itself
- The additional community developed plugins provide even more flexibility
- By combining Jenkins with Ant, Gradle, or other Build Automation tools, the possibilities are limitless

Why Jenkins? Free/OSS!

- Jenkins is released under the MIT License
- There is a large support community and thorough documentation
- It’s easy to write plugins
- Think something is wrong with it? You can fix it!

JENKINS

What can Jenkins do?

- Generate test reports
- Integrate with many different Version Control Systems
- Push to various artifact repositories
- Deploys directly to production or test environments
- Notify stakeholders of build status
- …and much more

How Jenkins works - Setup

- When setting up a project in Jenkins, out of the box you have the following general options:
  - Associating with a version control server
  - Triggering builds
  - Polling, Periodic, Building based on other projects
  - Execution of shell scripts, bash scripts, Ant targets, and Maven targets
  - Artifact archival
  - Publish JUnit test results and Javadocs
  - Email notifications
- As stated earlier, plugins expand the functionality even further

How Jenkins works - Building

- Once a project is successfully created in Jenkins, all future builds are automatic

Building...

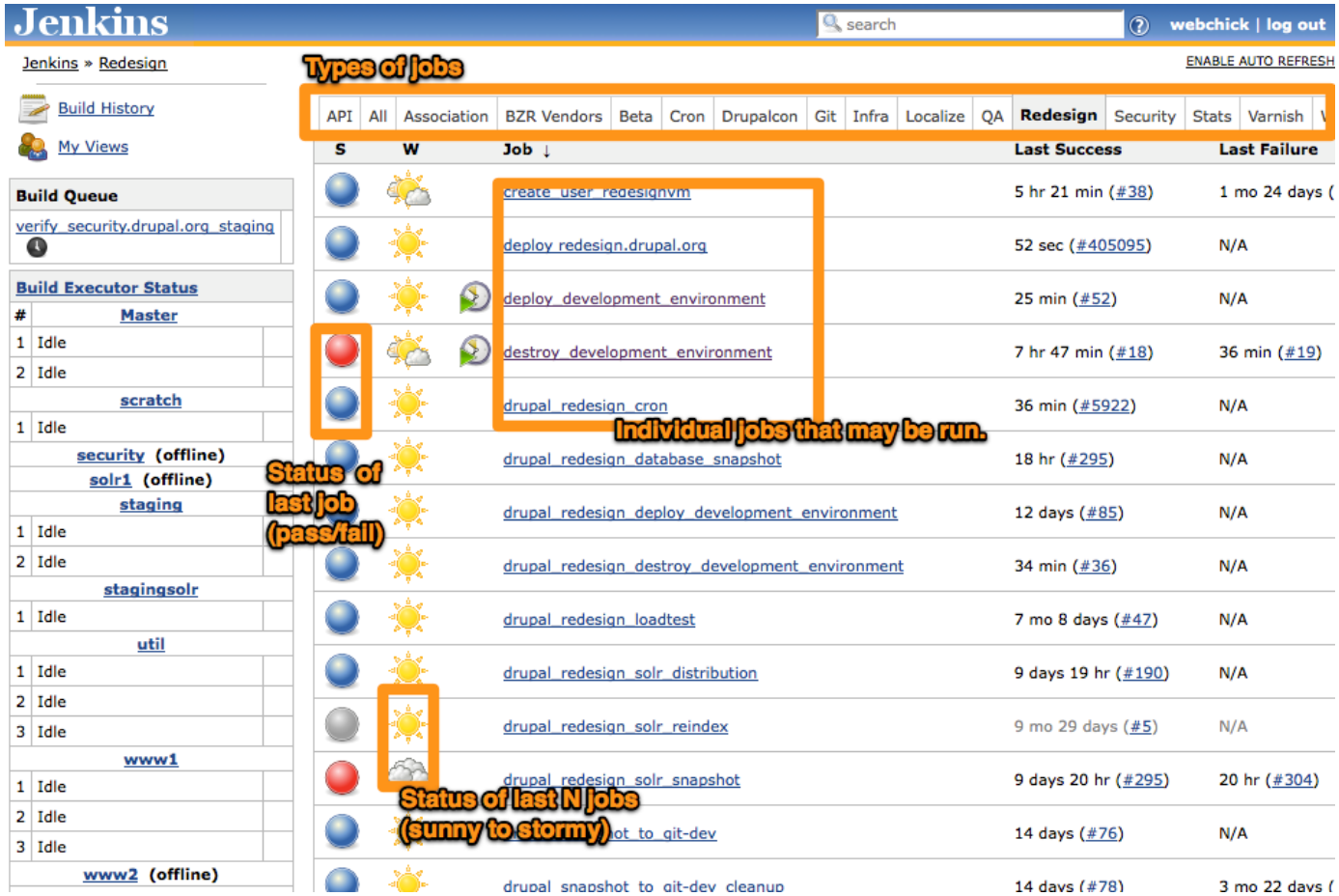
  - Jenkins executes the build in an executer
    - By default, Jenkins gives one executer per core on the build server
  - Jenkins also has the concept of Agent build servers
    - Useful for building on different architectures
    - Distribution of load

How Jenkins works - Reporting

- Jenkins comes with basic reporting features
  - Keeping track of build status
  - Last success and failure
  - “Weather” – Build trend
- These can be greatly enhanced with the use of pre-build plugins
  - Unit test coverage
  - Test result trending
  - Findbugs, Checkstyle, PMD

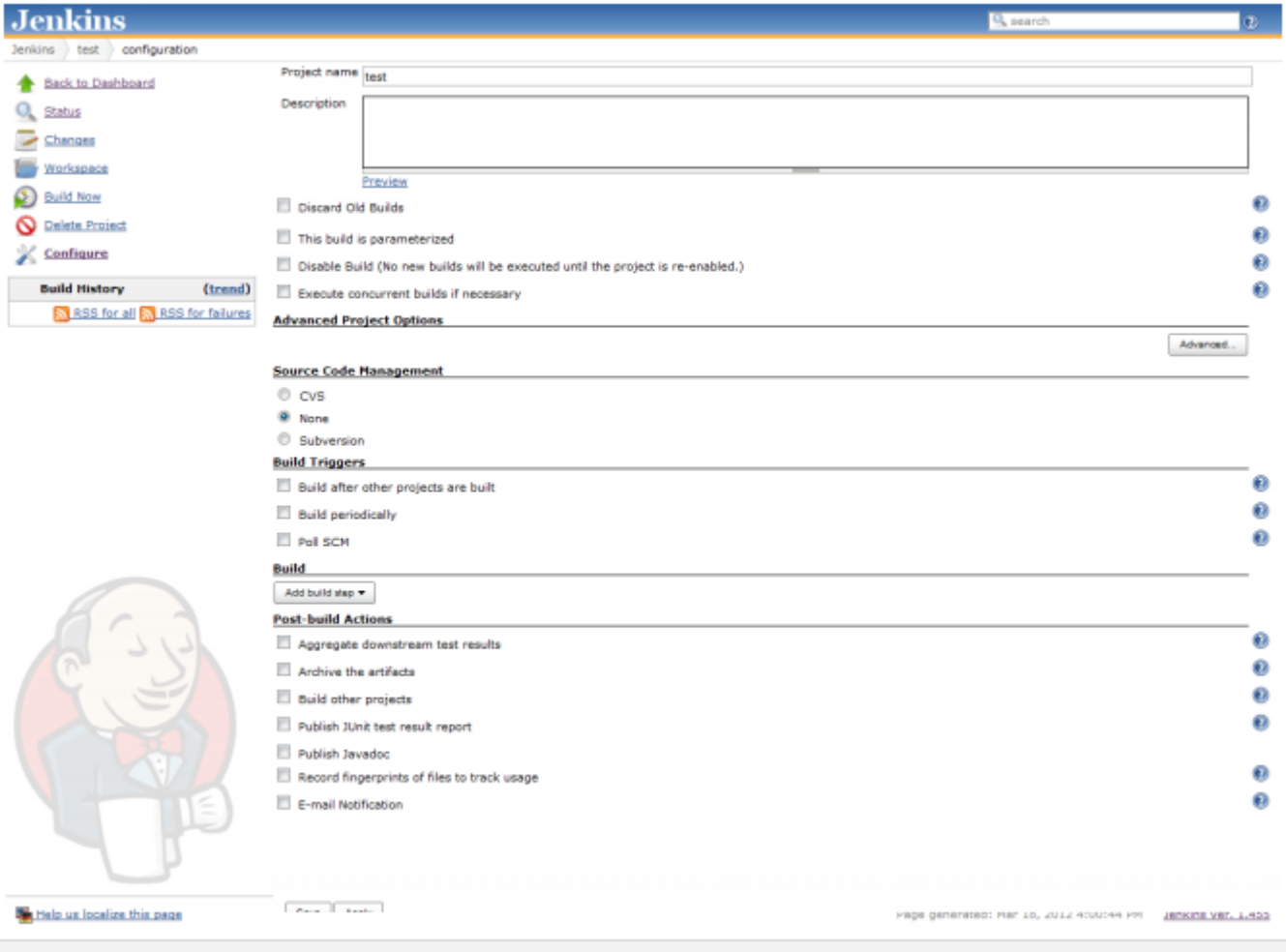
JENKINS

Jenkins by example – Project Status



JENKINS

Jenkins by example – New Project



Enhancing Jenkins

Jenkins plugin system can enable a wide range of features including (but certainly not limited to)

- **SCM**
  - Mercurial, Git, Subversion
- **Testing**
  - Selenium, Windmill, TestLink
- **Notifications**
  - IRC,Twitter, Jabber
- **Reporting**
  - Doxygen, PMD, Findbugs
- **Artifact Saving**
  - Artifactory, Amazon S3, SCP
- **Triggers**
  - Jabber, Directory Watchers
- **External Integration**
  - GitHub, Bugzilla, JIRA
- **And most importantly – The CI Game**
  - A points based game where developers compete against each other to develop the most stable, well- tested code

JENKINS

Who uses Jenkins?

Running Jenkins yourself

- Jenkins is packaged as a WAR, so you can drop it into whichever servlet container you prefer to use
- Jenkins comes pre-packaged with a servlet if you just want a light- weight implementation
- Native/Supported packages exist for
  - Windows
  - Ubuntu/Debian
  - Redhat/Fedora/CentOS
  - Mac OSX
  - OpenSUSE
  - FreeBSD
  - OpenBSD
  - Solaris/OpenIndiana
  - Gentoo

Running Jenkins yourself – Updates

- Jenkins has two release lines
  - Standard releases
  - Weekly bug fixes and features
- Long-Term Support releases
  - Updates about every 3 months
  - Uses a “Stable but older” version from the standard release line
  - Changes are limited to backported, well-tested modifications

Letting someone else run Jenkins

- There are also cloud-based solutions that can provide a Jenkins instance
  - Cloudbees - http://www.cloudbees.com/
  - ShiningPanda - https://www.shiningpanda.com/

Tying it into Agile

- For an Agile team, Jenkins provides everything needed for a robust continuous build system
- Jenkins supports Agile principles by constantly providing access to working copies of software
- Jenkins' extensibility allows the system to adapt to many different pre-existing environments

Putting it all together

- While an integral part of a CI system, Jenkins is by no means the only component
- In order for a CI system to function, a common repository for the codebase needs to exist
- A database of artifacts needs to exist, so deliveries can be made at past iterations
- The last step in a CI process is the deployment of the components built
- …and none of this matters if the developers don't use the system; procedures need to ensure the system is used as intended

Conclusion

- Continuous integration is a necessity on complex projects due to the benefits it provides regarding early detection of problems

- A good continuous build system should be flexible enough to fit into pre-existing development environments and provide all the features a team expects from such a system

- Jenkins, a continuous build system, can be an integral part of any continuous integration system due to it’s core feature set and extensibility through a plugin system

References

- Continuous Integration – Martin Fowler
  http://www.martinfowler.com/articles/continuousIntegration.html
- Hudson
  http://hudson-ci.org/
- Hudson Continuous Integration Server
  http://www.code-magazine.com/articleprint.aspx?quickid=0906071&printmode=true
- The Hudson Book
  http://www.eclipse.org/hudson/the-hudson-book/book-hudson.pdf
- Jenkins
  https://wiki.jenkins-ci.org
- Monkey Image
  http://corrines-corner2006.blogspot.com/2011/09/freebie-monday_26.html
- What is Continuous Integration
  http://confluence.public.thoughtworks.org/display/CCNET/What+is+Continuous+Integration

JENKINS

Why Jenkins? Award winning!

- InfoWorld Bossies Award, 2011

- O'Reilly Open-Source Award, 2011

- ALM&SCM, SDTimes 100, 2010, 2011

- GlassFish Community Innovation Award 2008

- Duke's Choice Award 2008


JENKINS


thank
you!