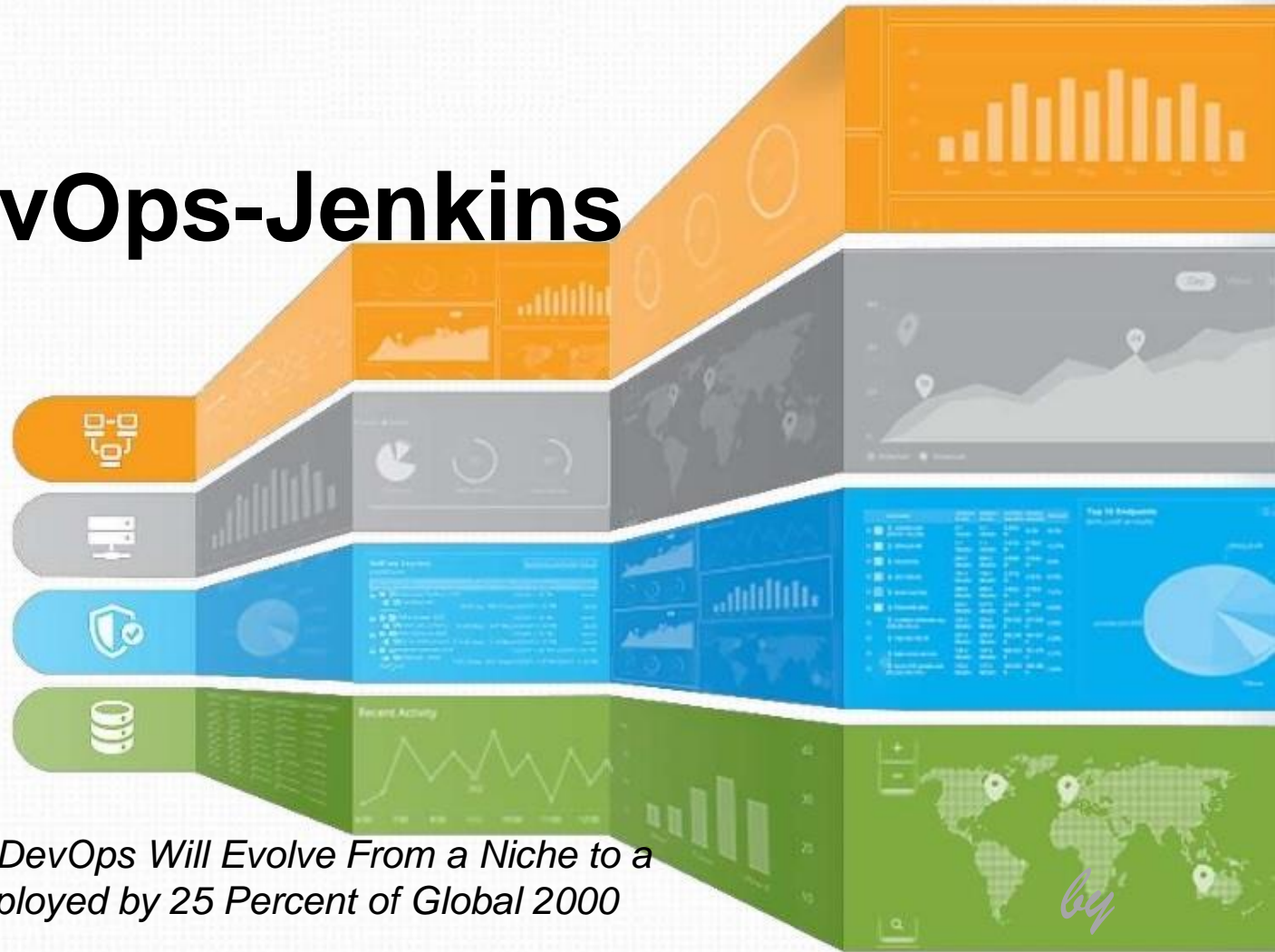


DevOps-Jenkins



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

JENKINS



Jenkins

“Continuous Build System”

JENKINS



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

JENKINS



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

JENKINS



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**

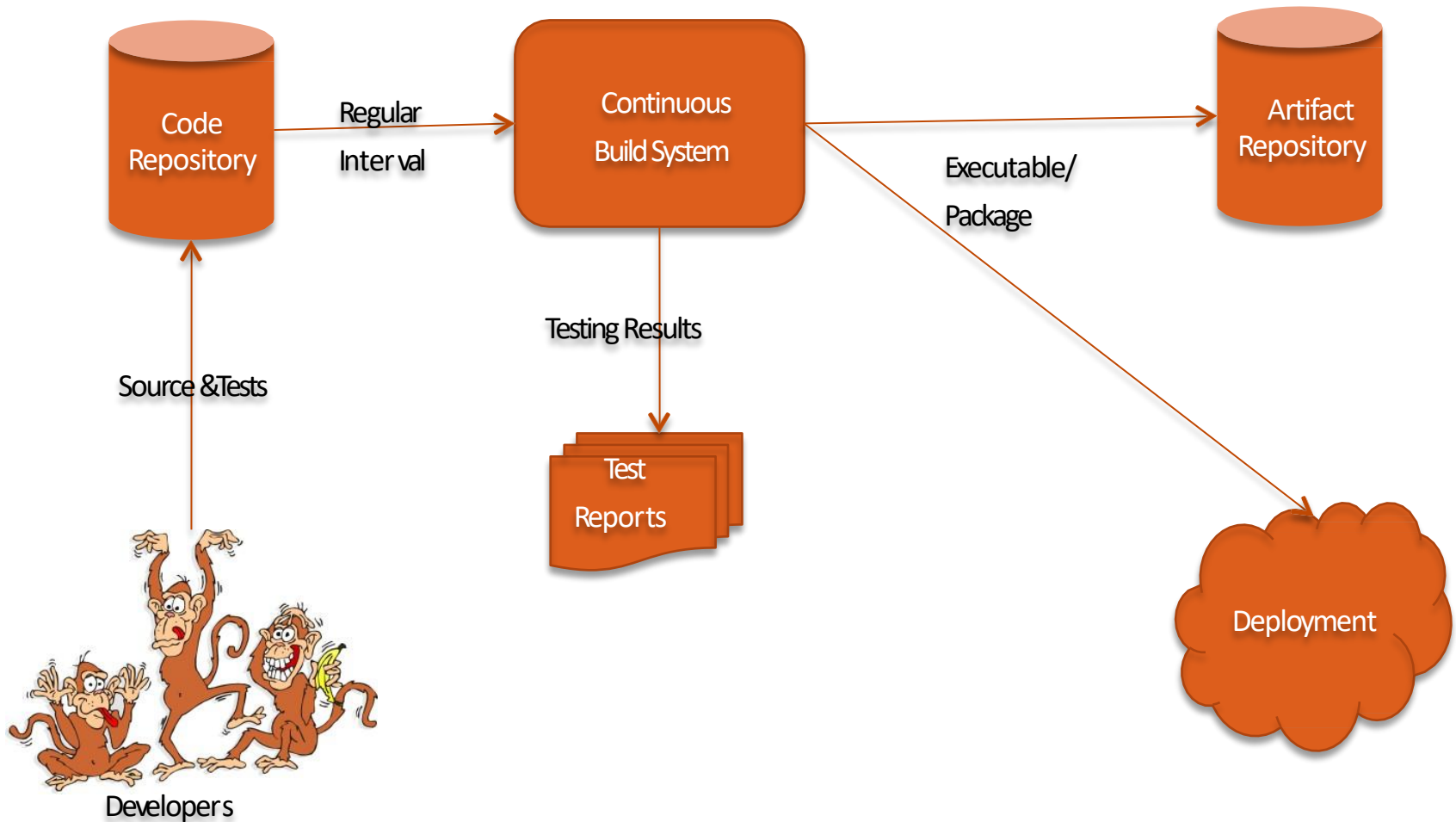
Versioned and stored so it can be distributed as is, if desired

- ☐ **Deployed**

Loaded onto a system where the developers can interact with it

JENKINS

CI - Workflow



JENKINS



CI – Benefits

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

JENKINS



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

JENKINS



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

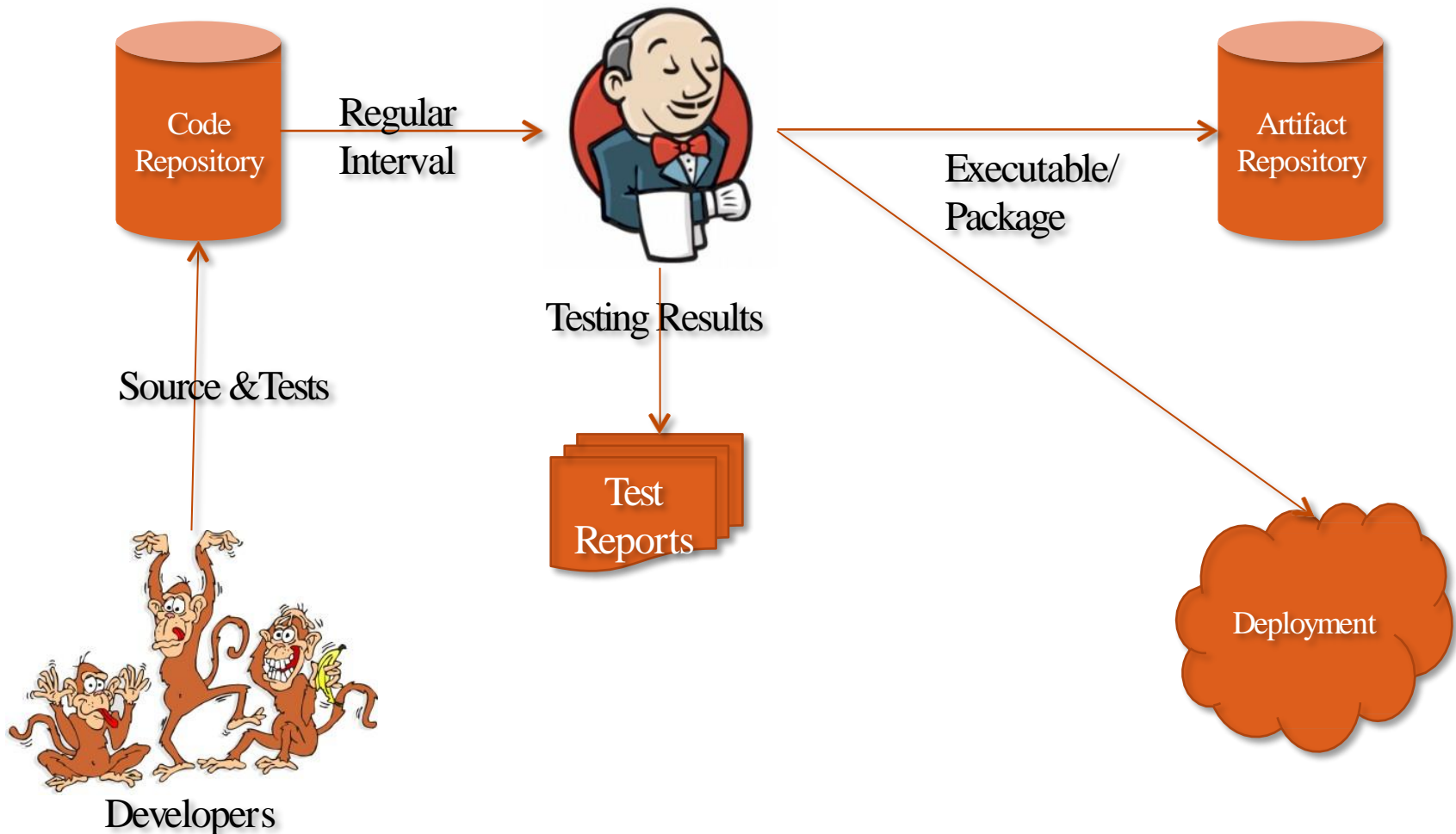


Jenkins - History

- ❑ 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



JENKINS



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

JENKINS



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

JENKINS



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 [webchick](#) | [log out](#) [ENABLE AUTO REFRESH](#)

Jenkins » [Redesign](#)

[Build History](#)
[My Views](#)

Build Queue
[verify_security.drupal.org_staging](#)

Build Executor Status

#	Master
1	Idle
2	Idle

[scratch](#)

1	Idle
---	------

[security](#) (offline)
[solr1](#) (offline)
[staging](#)

1	Idle
2	Idle

[stagingsolr](#)

1	Idle
---	------

[util](#)

1	Idle
2	Idle
3	Idle

[www1](#)

1	Idle
2	Idle
3	Idle

[www2](#) (offline)

Types of Jobs

API	All	Association	BZR Vendors	Beta	Cron	Drupalcon	Git	Infra	Localize	QA	Redesign	Security	Stats	Varnish
S	W	Job ↓	Last Success	Last Failure										
		create_user_redesignvm	5 hr 21 min (#38)	1 mo 24 days (
		deploy_redesign.drupal.org	52 sec (#405095)	N/A										
		deploy_development_environment	25 min (#52)	N/A										
		destroy_development_environment	7 hr 47 min (#18)	36 min (#19)										
		drupal_redesign_cron	36 min (#5922)	N/A										
		drupal_redesign_database_snapshot	18 hr (#295)	N/A										
		drupal_redesign_deploy_development_environment	12 days (#85)	N/A										
		drupal_redesign_destroy_development_environment	34 min (#36)	N/A										
		drupal_redesign_loadtest	7 mo 8 days (#47)	N/A										
		drupal_redesign_solr_distribution	9 days 19 hr (#190)	N/A										
		drupal_redesign_solr_reindex	9 mo 29 days (#5)	N/A										
		drupal_redesign_solr_snapshot	9 days 20 hr (#295)	20 hr (#304)										
		not_to_git-dev	14 days (#76)	N/A										
		drupal_snapshot_to_git-dev_cleanup	14 days (#78)	3 mo 22 days (

Individual jobs that may be run.

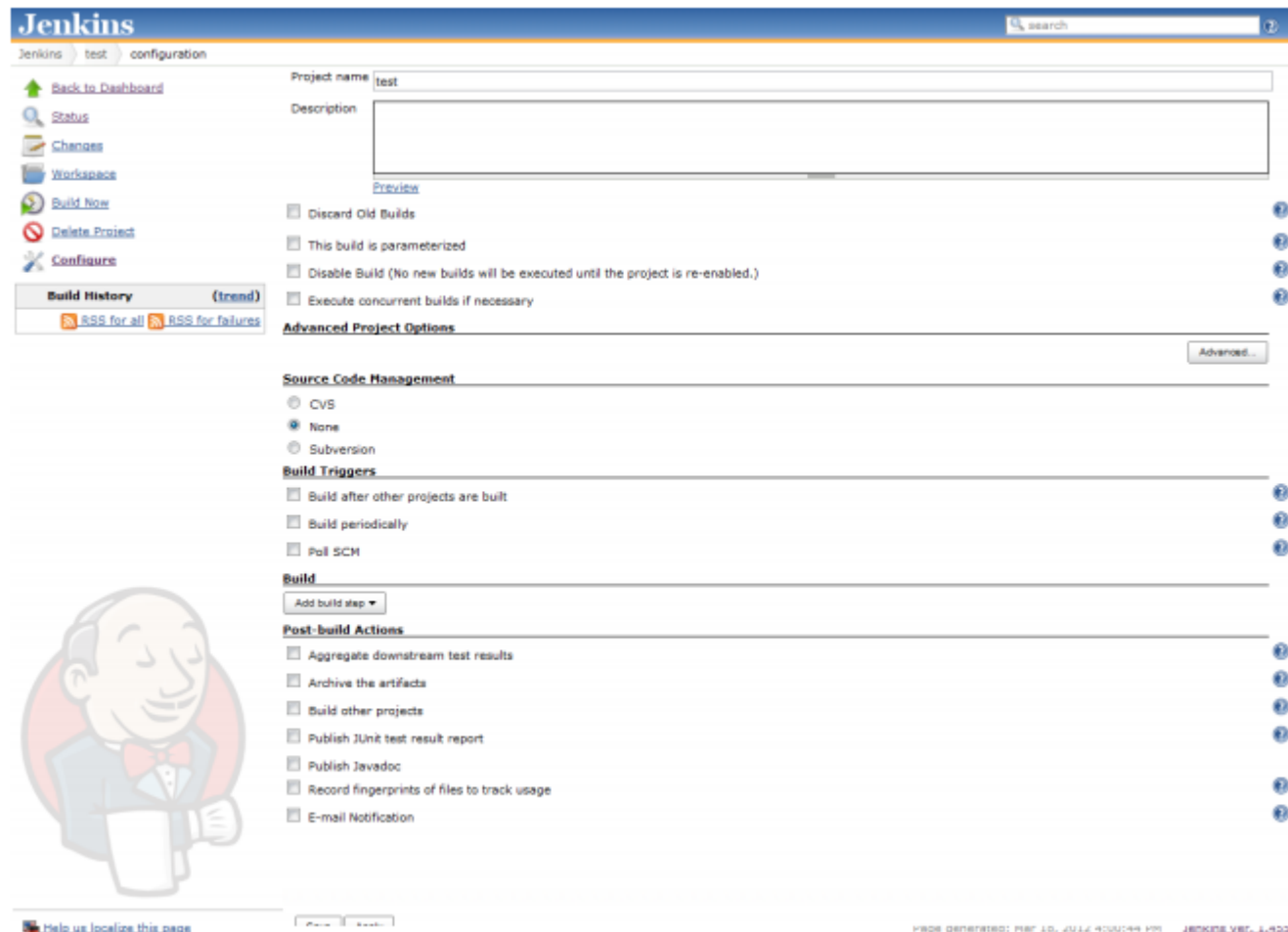
Status of last job (pass/fail)

Status of last N jobs (sunny to stormy)

JENKINS



Jenkins by example – New Project



The screenshot shows the Jenkins 'New Project' configuration page. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, and Configure. Below these is a 'Build History' section with a '(trend)' link and two RSS feeds. The main content area is titled 'Jenkins test configuration' and includes a search bar. The 'Project name' is set to 'test'. The 'Description' field is empty. Below the description is a 'Preview' link. The 'Advanced Project Options' section includes checkboxes for 'Discard Old Builds', 'This build is parameterized', 'Disable Build', and 'Execute concurrent builds if necessary'. The 'Source Code Management' section has radio buttons for 'CVS', 'None' (selected), and 'Subversion'. The 'Build Triggers' section has checkboxes for 'Build after other projects are built', 'Build periodically', and 'Poll SCM'. The 'Build' section has a dropdown menu for 'Add build step'. The 'Post-build Actions' section has checkboxes for 'Aggregate downstream test results', 'Archive the artifacts', 'Build other projects', 'Publish JUnit test result report', 'Publish Javadoc', 'Record fingerprints of files to track usage', and 'E-mail Notification'. A cartoon character of a man in a tuxedo is visible in the bottom left corner. The footer contains a 'Help us localize this page' link and a page generation timestamp.

Jenkins test configuration

Project name: test

Description:

Preview

☐ Discard Old Builds

☐ This build is parameterized

☐ Disable Build (No new builds will be executed until the project is re-enabled.)

☐ Execute concurrent builds if necessary

Advanced Project Options

Source Code Management

☐ CVS

☒ None

☐ Subversion

Build Triggers

☐ Build after other projects are built

☐ Build periodically

☐ Poll SCM

Build

Add build step

Post-build Actions

☐ Aggregate downstream test results

☐ Archive the artifacts

☐ Build other projects

☐ Publish JUnit test result report

☐ Publish Javadoc

☐ Record fingerprints of files to track usage

☐ E-mail Notification

Help us localize this page

Page generated: Mar 10, 2014 10:04:44 PM JENKINS VER. 1.523

JENKINS



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



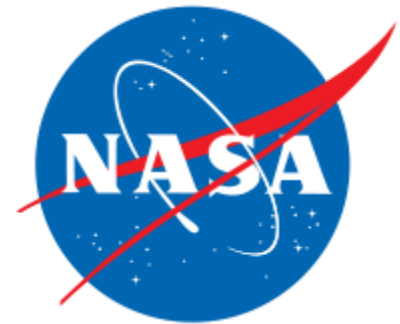
Who uses Jenkins?

LinkedIn

eBay

CARFAX[®]
VEHICLE HISTORY REPORTS

NETFLIX



BOEING

mozilla
FOUNDATION

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

JENKINS



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/>

JENKINS



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

JENKINS



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

JENKINS



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 its core feature set and extensibility through a plugin system

JENKINS



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!