Continuous Integration with Jenkins

Continuous Integration

- Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day.
- Each check-in is then verified by an automated build, allowing teams to detect problems early.
- By integrating regularly, you can detect errors quickly, and locate them more easily.
- Continuous Integration emerged in the Extreme Programming (XP) community, and XP advocates Martin Fowler and Kent Beck first wrote about continuous integration circa 1999.

Continuous Integration

- *“Continuous Integration doesn’t get rid of bugs, but it does make them dramatically easier to find and remove.”*
  - Martin Fowler
- Because you’re integrating so frequently, there is significantly less back-tracking to discover where things went wrong, so you can spend more time building features.
- Continuous Integration is cheap. Not continuously integrating is costly.
- If you don’t follow a continuous approach, you’ll have longer periods between integrations. This makes it exponentially more difficult to find and fix problems. Such integration problems can easily knock a project off-schedule, or cause it to fail altogether.

Why Continuous Integration?

- Integration is hard, effort increase exponentially with
  - Number of components
  - Number of bugs
  - Time since last integration

Continuous Integration benefits

- Say goodbye to long and tense integrations
- Increase visibility which enables greater communication
- Catch issues fast and nip them in the bud
- Spend less time debugging and more time adding features
- Proceed in the confidence you're building on a solid foundation
- Stop waiting to find out if your code's going to work
- Reduce integration problems allowing you to deliver software more rapidly

CI Practices

- Maintain a single source repository
  - Check in all the code in a Single Source repository
  - You can use any source code repository like GIT, SVN, PERFORCE, Clearcase etc
  - Check in Everything required for a build
    - test scripts,
    - properties files,
    - database schema,
    - install scripts, and
    - third party libraries.

CI Practices

- **Automate the Build**
- **Make Your Build Self-Testing**
- **Everyone Commits To the Mainline Every Day**
- **Every Commit Should Build the Mainline on an Integration Machine**
- **Fix Broken Builds Immediately**
- **Keep the Build Fast**
- **Test in a Clone of the Production Environment**
- **Make it Easy for Anyone to Get the Latest Executable**
- **Everyone can see what's happening**
- **Automate Deployment**

CI Process

- Developers check out code into their private workspaces.
- When done, commit the changes to the repository.
- The CI server monitors the repository and checks out changes when they occur.
- The CI server builds the system and runs unit and integration tests.
- The CI server releases deployable artefacts for testing.
- The CI server assigns a build label to the version of the code it just built.
- The CI server informs the team of the successful build.
- If the build or tests fail, the CI server alerts the team.
- The team fix the issue at the earliest opportunity.
- Continue to continually integrate and test throughout the project.

Teams Responsibility

- Check in frequently
- Don’t check in broken code
- Don’t check in untested code
- Don’t check in when the build is broken
- Don’t go home after checking in until the system builds

Continuous Deployment

- **Continuous delivery** (**CD**) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.
- It aims at building, testing, and releasing software faster and more frequently.
- The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production.
- A straightforward and repeatable deployment process is important for continuous delivery.
- Continuous Deployment is closely related to Continuous Integration and refers to the release into production of software that passes the automated tests.

CI/CD Tools

- Jenkins
- Travis
- Bamboo

Build

Build is the process of integrating, building and compiling the software that is produced.

For large software teams, there is generally a continuous integration process where software developers check-in their changes, and if anyone breaks the build, it needs to be addressed.

The integration process can take a lot of processing power, and several hours. This infrastructure needs to be monitored and optimized.

Commonly used Build tools

- Maven
- Bundle
- Ant
- Build master
- Quick Build

Build

Build is the process of integrating, building and compiling the software that is produced.

For large software teams, there is generally a continuous integration process where software developers check-in their changes, and if anyone breaks the build, it needs to be addressed.

The integration process can take a lot of processing power, and several hours. This infrastructure needs to be monitored and optimized.

Commonly used Build tools

- Maven
- Bundle
- Ant
- Build master
- Quick Build

Quality Assurence

- **QA owns continuous improvement and quality tracking** across the entire development cycle. They are the ones who are primarily responsible for identifying problems not just in the product but also in the process, and recommending changes wherever they can.
- **Tests are code**, as any test automation expert will tell you. It's a necessity, of course. If your process is designed to publish a new release every day (or every hour) there is no room for manual testing. You must develop automation systems, through code, that can ensure quality standards are maintained.
- **Automation rules**. Anything that can be automated, should be automated. When Carl describes Unbounce's deployment process as "push-button easy," this is what he's talking about.
- **Testers are the quality advocates**, influencing both development and operational processes. They don't just find bugs. They look for any opportunity to improve repeatability and predictability.

QA Tools

- Code testing tools
  - Sonar
  - Fortify
  - Coverty
- Automated Testing tools
  - Selinium
  - Sause
  - Cucumber
  - Junit

Continuous Monitoring

- DevOps teams then face significant challenges in guaranteeing expected application behavior:
  - Understanding application performance before and after new code is pushed and pinpointing defects early, before they spread.
  - Diving back into the history of deployments, determining the impact on the infrastructure throughput and response time.
  - Forecasting infrastructure utilization bottlenecks due to changes into the code or variations in the workload.
  - Rather than waiting for production performance data to analyze what went wrong, the DevOps team is able to develop performance analytics models that can anticipate operational and quality problems before the delivery phase.

Continuous Monitoring Tools

- Infrastructure Monitoring
  - **Nagios, Zabbix & Sensu**
- Iaas Monitoring
  - AWS Cloud watch, Openstack Celiometer, Stack driver
- Application Performance Monitoring
  - New Relic and App Dynamics

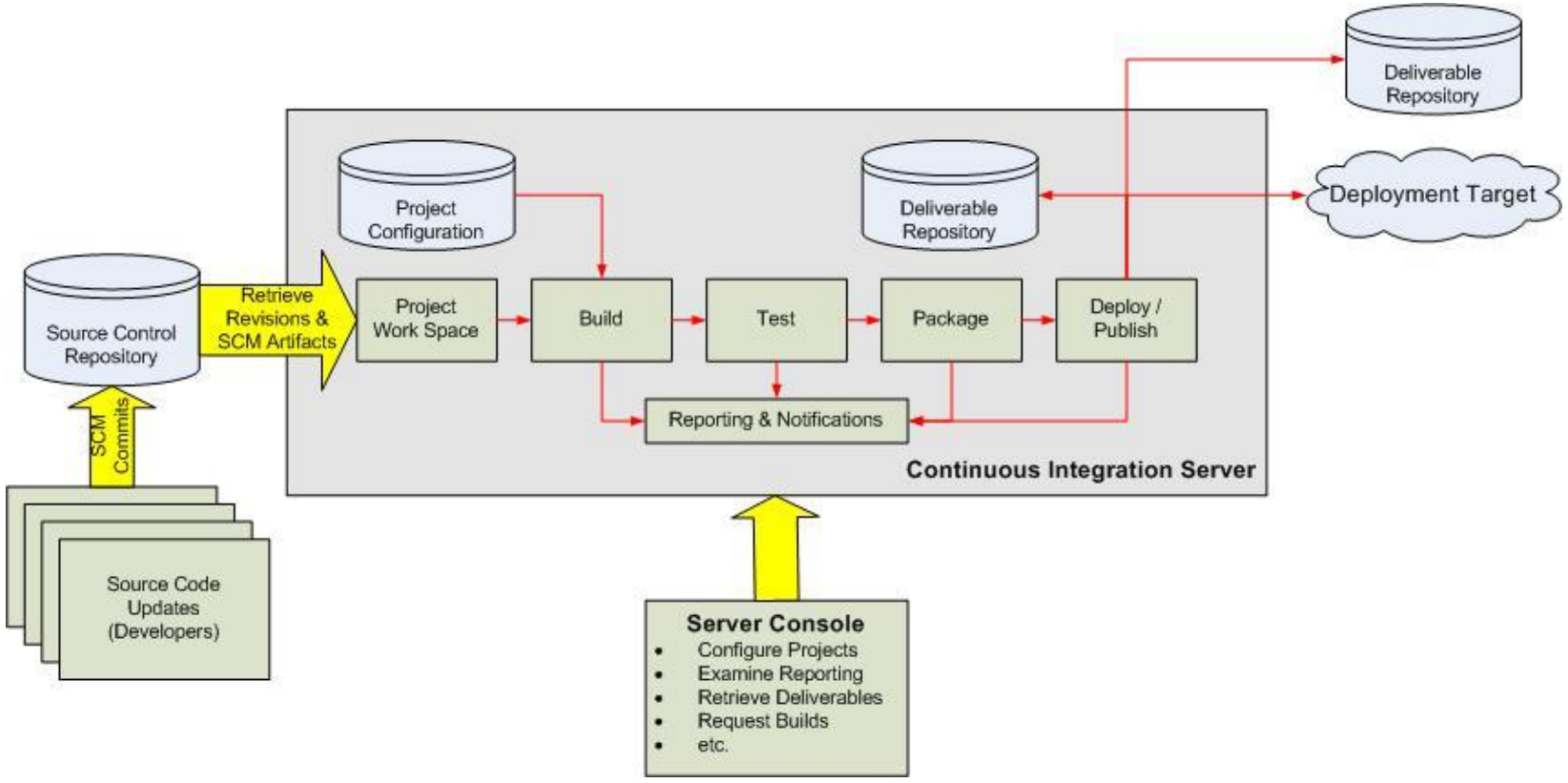
What’s Jenkins

1. An open source CI server
2. More then 23000 installations (Jul 2010)
3. Plug-in extensibility (Over 370 plugins)
4. MIT license

What’s Jenkins/Hudson

1. An open source CI server
2. More then 23000 installations (Jul 2010)
3. Plug-in extensibility (Over 370 plugins)
4. MIT license

Continuous Integration Overview



Ref: http://www.javaworld.com/javaworld/jw-12-2008/images/CIOverview.jpg

Jenkins Features

- Trigger a build
- Get source code from repository
- Automatically build and test
- Generate report & notify
- Deploy
- Distributed build

Jenkins Requirement

- Web Server (Tomcat, WebLogic, …)
- Build tool (Maven, Ant)
- SCM (Git, Svn, Cvs, …)
- Automation Deployment (Docker, Shell scripts, etc)

Jenkins Plugins

- Build triggers
- Source code management
- Build tools
- Build wrappers
- Build notifiers
- Build reports
- Artifact uploaders
- UI plugins
- Authentication and user management

Build Trigger

- Manually click build button
- Build periodically
- Build whenever a SNAPSHOT dependency is built
- Build after other projects are built
- Poll SCM
- IRC, Jabber, …

Get Source Code (1/2)

- CVS (build-in)
- SVN (build-in)
- GIT (requires Git)
- ClearCase (requires ClearCase)
- Mercurial, PVCS, VSS, …

Code Change History

Build Tools

- Java
  - Maven (build-in), Ant, Gradle
- .Net
  - MSBuild, PowerShell
- Shell script
  - Python, Ruby, Groovy

Build Wrapper

- Build name (version no) setter
- Virtual machine (VMWare, Virtual Box)
- Set environment variable
- ClearCase release plugin
- …

Build Notifier

- E-mail
- Twitter
- Jabber
- IRC
- RSS
- Google calendar
- …

Build Report

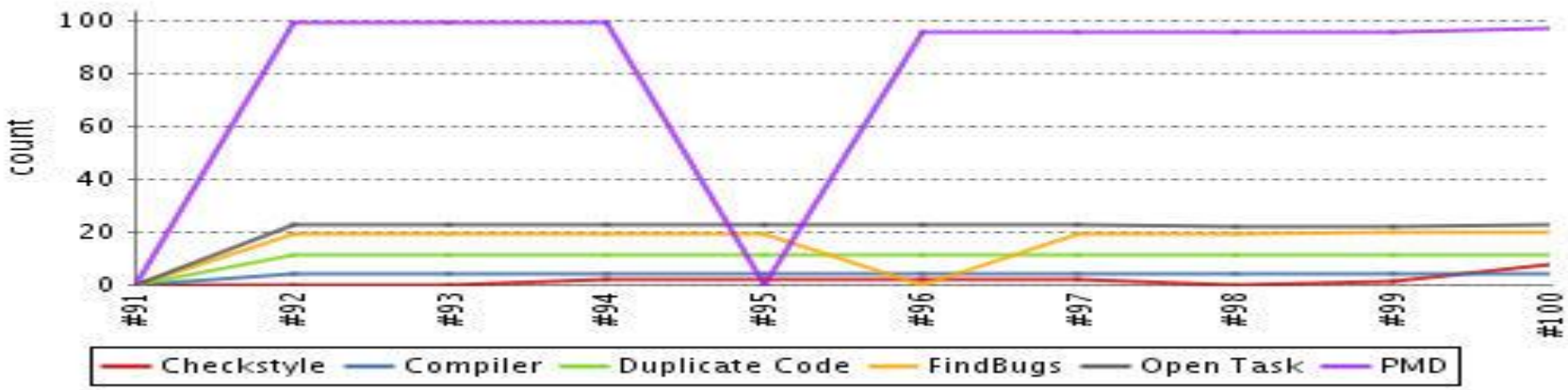
- Static Code Analysis
  - Checkstyle, PMD, Findbugs, Compiler Warning
- Test Report & Code Coverage
  - JUnit, TestNG, Cobertura, Clover
- Open Tasks

Static Code Analysis

All **DashBoard** +

Warnings per Job

Job ↓	Checkstyle	Duplicate Code	FindBugs	PMD	Open Tasks	Compiler Warnings	Total
M-Multi-Freestyle	14	-	0	86	24	0	124
M-Multi-Maven	14	11	12	80	0	4	121
M-Single-Freestyle	0	0	15	18	99	3	135
M-Single-Freestyle-Site	5967	2	-	18	5	0	5992
M-Single-Maven	-	-	-	-	121	0	121
M-Single-Maven-2	-	4	16	243	4	0	267
M-Single-Maven-Site	1784	2	72	2	0	-	1860



CheckStyle

CheckStyle Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
6766	2	0

Summary

Total	High Priority	Normal Priority	Low Priority
6766	6766	0	0

Details

Modules | Package | Files | Categories | **Types** | Warnings | Details | New

Type	Total	Distribution
AvoidInlineConditionalsCheck	47	
AvoidNestedBlocksCheck	2	
AvoidStarImportCheck	1	
ConstantNameCheck	11	
DesignForExtensionCheck	789	
EmptyBlockCheck	29	
FinalClassCheck	5	
FinalParametersCheck	980	
HiddenFieldCheck	288	
HideUtilityClassConstructorCheck	9	
InnerAssignmentCheck	1	
InterfaceIsTypeCheck	8	
JavadocMethodCheck	1472	
JavadocPackageCheck	1	

FindBugs

FindBugs Result

Warnings Trend

All Warnings	New this build	Fixed Warnings
22	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
22	0	0	22

Details

Modules | Package | Files | Categories | **Types** | Warnings | Details

Type	Total	Distribution
CN_IMPLEMENTS_CLONE_BUT_NOT_CLONEABLE	1	
DLS_DEAD_LOCAL_STORE	1	
EI_EXPOSE_REP	4	
EI_EXPOSE_REP2	7	
EQ_CHECK_FOR_OPERAND_NOT_COMPATIBLE_WITH_THIS	3	
EQ_DOESNT_OVERRIDE_EQUALS	2	
ICAST_IDIV_CAST_TO_DOUBLE	1	
OBL_UNSATISFIED_OBLIGATION	1	
RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	1	
WA_AWAIT_NOT_IN_LOOP	1	
Total	**22**	

Open Tag

Open Tasks

Open Tasks Trend

All Open Tasks	New Tasks	Fixed Tasks
19	19	0

Summary

Total	High Priority	Normal Priority
19	0	19

Details

Package | Files | **Warnings** | Details | New

File	Package	Line	Priority	Type	Category
AbstractClearCaseScm.java	hudson.plugins.clearcase	255	Normal	TODO	
AbstractClearCaseScm.java	hudson.plugins.clearcase	256	Normal	TODO	
AbstractClearCaseScm.java	hudson.plugins.clearcase	257	Normal	TODO	
ClearTool.java	hudson.plugins.clearcase	196	Normal	TODO	
ClearTool.java	hudson.plugins.clearcase	237	Normal	TODO	
ClearTool.java	hudson.plugins.clearcase	238	Normal	TODO	

Duplicate Code

Duplicate Code Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
16	0	0

Summary

Total	High Priority	Normal Priority	Low Priority
16	0	10	6

Details

Package | Files | **Warnings** | Details | Normal | Low

File	Number of lines	Duplicated in
NetscapeDraftSpec.java:120	17	BestMatchSpec.java:117
BestMatchSpec.java:117	17	NetscapeDraftSpec.java:120
SSLSocketFactory.java:462	28	PlainSocketFactory.java:154
AbstractAuthenticationHandler.java:82	23	AuthSchemeBase.java:88
RequestProxyAuthentication.java:95	28	RequestTargetAuthentication.java:86
RFC2109DomainHandler.java:47	32	BasicDomainHandler.java:45
AuthSchemeBase.java:88	23	AbstractAuthenticationHandler.java:82
NetscapeDraftSpec.java:137	19	BrowserCompatSpec.java:163
PlainSocketFactory.java:154	28	SSLSocketFactory.java:462
RequestTargetAuthentication.java:86	28	RequestProxyAuthentication.java:95
BrowserCompatSpec.java:163	19	NetscapeDraftSpec.java:137
DefaultRedirectHandler.java:141	44	DefaultRedirectStrategy.java:133
BrowserCompatSpec.java:128	34	BestMatchSpec.java:101
BasicDomainHandler.java:45	32	RFC2109DomainHandler.java:47
DefaultRedirectStrategy.java:133	44	DefaultRedirectHandler.java:141
BestMatchSpec.java:101	34	BrowserCompatSpec.java:128

Test Report

Tests							
Job	Success		Failed		Skipped		Total
	#	%	#	%	#	%	#
Eden-ActiveWorlds	127	100%	0	0%	0	0%	127
Eden-Eden	266	100%	0	0%	0	0%	266
Eden-MondiDinamiciWebservice	155	100%	0	0%	0	0%	155
Eden-Palm	20	100%	0	0%	0	0%	20
Total	568	100%	0	0%	0	0%	568

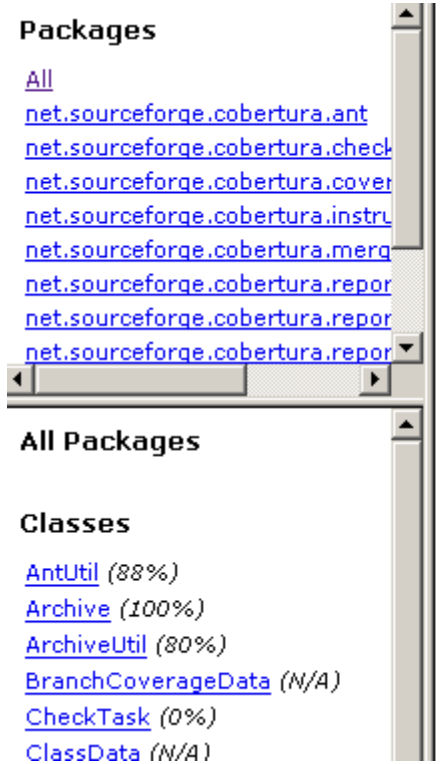
Test Result

0 failures (±0) , 2 skipped (±0)

1,513 tests (±0)

Module	Fail	(diff)	Total	(diff)
org.apache.httpcomponents:httpclient	0		582	
org.apache.httpcomponents:httpclient-cache	0		920	
org.apache.httpcomponents:httpmime	0		11	

Test Code Coverage



Coverage Report - All Packages

Package	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	55	75%	1625/2179	64%	472/738	2.319
net.sourceforge.cobertura.ant	11	52%	170/330	43%	40/94	1.848
net.sourceforge.cobertura.check	3	0%	0/150	0%	0/76	2.429
net.sourceforge.cobertura.coveragedata	13	N/A	N/A	N/A	N/A	2.277
net.sourceforge.cobertura.instrument	10	90%	460/510	75%	123/164	1.854
net.sourceforge.cobertura.merge	1	86%	30/35	88%	14/16	5.5
net.sourceforge.cobertura.reporting	3	87%	116/134	80%	43/54	2.882
net.sourceforge.cobertura.reporting.html	4	91%	475/523	77%	156/202	4.444
net.sourceforge.cobertura.reporting.html.files	1	87%	39/45	62%	5/8	4.5
net.sourceforge.cobertura.reporting.xml	1	100%	155/155	95%	21/22	1.524
net.sourceforge.cobertura.util	9	60%	175/291	69%	70/102	2.892
someotherpackage	1	83%	5/6	N/A	N/A	1.2

Report generated by Cobertura 1.9 on 6/9/07 12:37 AM.

Ref: http://cobertura.sourceforge.net/sample/

Artifact uploaders

- Tomcat
- JBoss
- Glassfish
- WebSphere
- FTP
- SSH

UI Enhancement

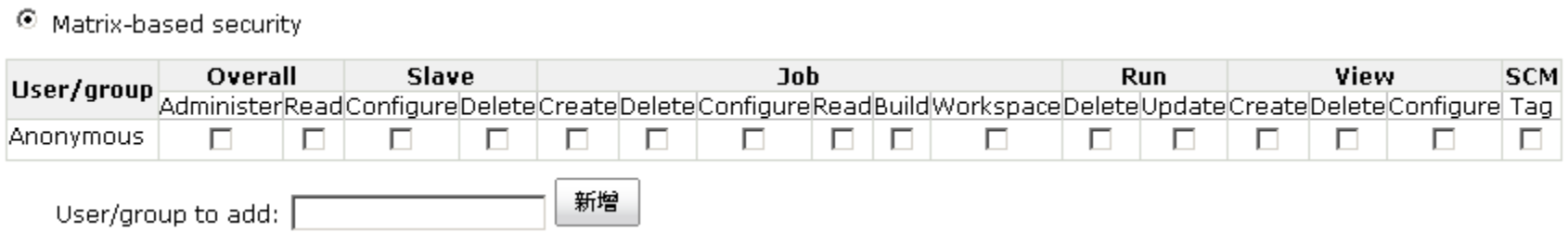
- Dashboard
- Sectioned view
- iPhone/Android

Security Management

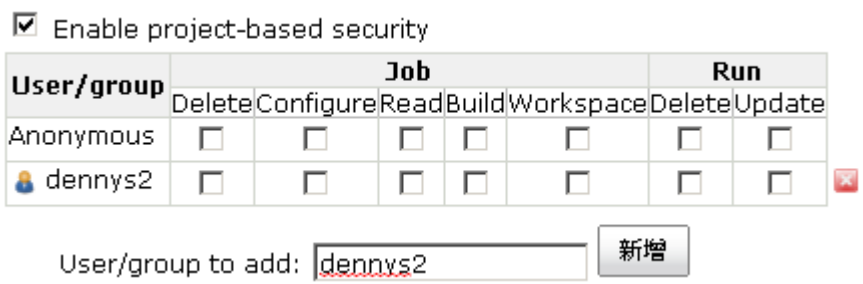
- Security Realm
  - LDAP
  - Jenkins's own user database
  - Delegate to servlet container
- Authorization
  - Anyone can do anything
  - Logged-in users can do anything
  - Matrix-based security
  - Project-based Matrix Authorization Strategy
  - Legacy mode

Security Management

- Matrix-based security

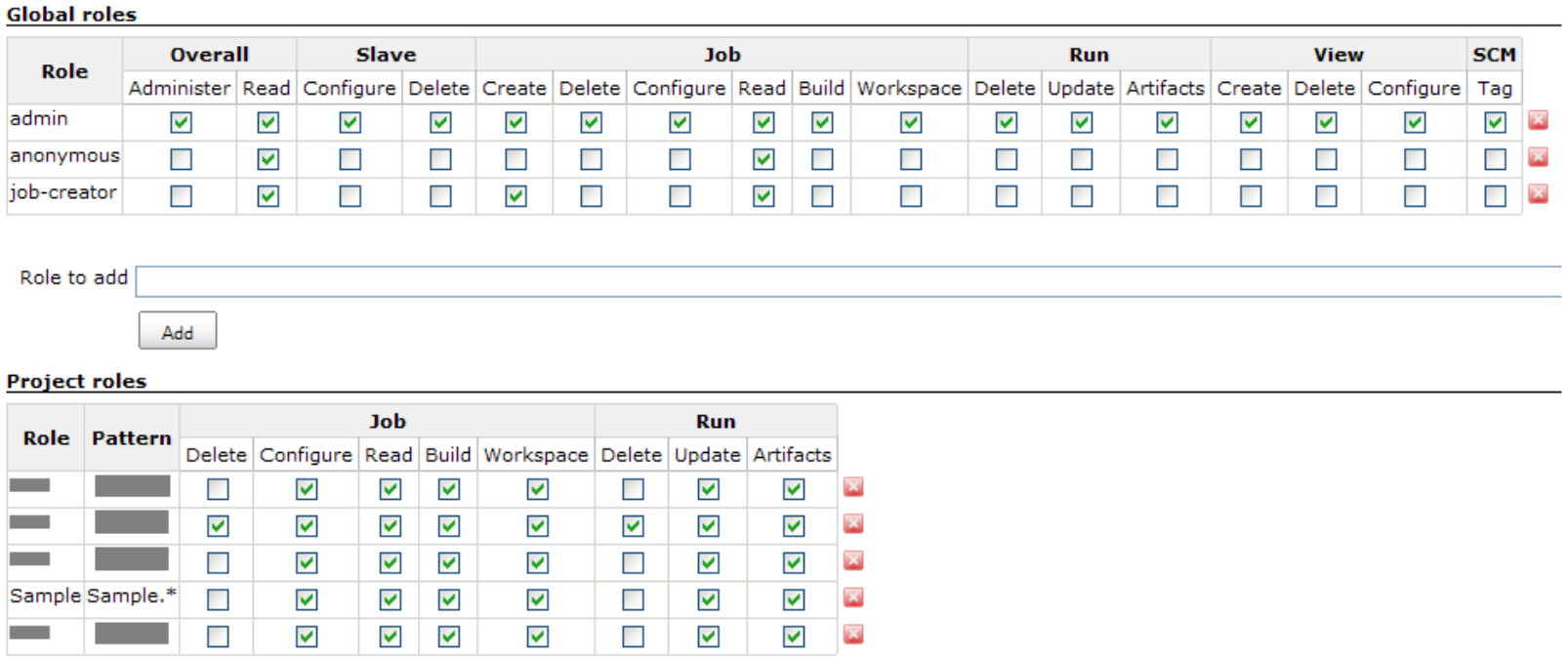


- Project-based Matrix Authorization



Security Management Plugins

- Active directory, OpenID, MySQL, …
- Role based privilege control



Global roles

Role	Overall		Slave		Job						Run			View			SCM
	Administer	Read	Configure	Delete	Create	Delete	Configure	Read	Build	Workspace	Delete	Update	Artifacts	Create	Delete	Configure	Tag
admin	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
anonymous	☐	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐
job-creator	☐	☑	☐	☐	☑	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐

Role to add

Add

Project roles

Role	Pattern	Job					Run		
		Delete	Configure	Read	Build	Workspace	Delete	Update	Artifacts
[redacted]	[redacted]	☐	☑	☑	☑	☑	☐	☑	☑
[redacted]	[redacted]	☑	☑	☑	☑	☑	☑	☑	☑
[redacted]	[redacted]	☐	☑	☑	☑	☑	☐	☑	☑
Sample	Sample.*	☐	☑	☑	☑	☑	☐	☑	☑
[redacted]	[redacted]	☐	☑	☑	☑	☑	☐	☑	☑

Plugin Usage Statistics

	A	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	plugin	2009-06	2009-07	2009-08	2009-09	2009-10	2009-11	2009-12	2010-01	2010-02	2010-03	2010-04	2010-05	2010-06	2010-07
2	TOTAL INSTALLS	11026	11918	12092	12972	13379	15222	15812	16578	17936	20037	20670	21210	22182	23159
3	ssh-slaves	7715	9306	9985	11081	11759	13668	14415	15302	16712	18832	19515	20059	21054	22040
4	maven-plugin	8568	9983	10483	11459	12076	13941	14617	15424	16762	18823	19462	19981	20937	21917
5	subversion	1791	6391	8035	9505	10614	12653	13563	14551	16071	18289	19035	19711	20771	21792
6	cvs	0	0	0	1	0	0	8	4625	9192	12384	13816	15173	16568	18001
7	scis-ad	0	0	2818	6634	8572	10848	11957	13133	14831	17164	18022	16502	14433	13328
8	analysis-c	0	0	0	1	1	2	1261	2449	3551	4454	4958	5446	5863	6338
9	email-ext	2349	2572	2694	2880	2956	3372	3492	3725	3965	4332	4473	4615	4906	5187
10	greenballs	1183	1417	1508	1700	1853	2198	2460	2698	2978	3376	3536	3708	3949	4248
11	findbugs	2781	2937	2973	3154	3107	3315	3302	3277	3554	3825	3843	3949	4040	4186
12	checkstyle	2354	2534	2588	2781	2728	2996	2914	2951	3228	3484	3595	3654	3754	3908
13	cobertura	2051	2185	2265	2431	2455	2713	2821	2911	3176	3495	3581	3658	3763	3906
14	warnings	2078	2225	2290	2338	2450	2676	2663	2651	2829	3061	3201	3249	3400	3498
15	pmd	2033	2189	2233	2422	2393	2572	2546	2578	2800	3098	3071	3160	3238	3402
16	disk-usage	1666	1803	1886	2026	2080	2369	2489	2535	2663	2928	2979	2980	3068	3169
17	sonar	1024	1137	1178	1332	1454	1730	1798	1869	2019	2368	2549	2641	2875	2944
18	tasks	1868	2004	1998	2055	2085	2329	2305	2306	2387	2600	2692	2662	2780	2833
19	violations	1366	1471	1531	1624	1720	1881	1947	2009	2200	2397	2407	2461	2523	2625
20	deploy	1221	1259	1292	1380	1478	1669	1727	1776	1862	2139	2213	2188	2374	2494
21	jira	1179	1255	1278	1387	1456	1628	1717	1775	1899	2153	2207	2245	2381	2489
22	chucknorri	1	1	1	168	688	985	1203	1419	1678	1972	2137	2262	2344	2476
23	svn-tag	1175	1312	1328	1382	1423	1643	1685	1716	1810	2034	2040	2084	2206	2338
24	git	558	609	661	693	798	938	1046	1159	1325	1613	1778	1853	1951	2265
25	active-directory	1123	1219	1237	1314	1365	1491	1571	1651	1790	1917	1989	2047	2095	2223

Ref: http://jenkins-ci.org/content/updated-usage-stats-available

Demo