

Обзор системы непрерывной интеграции Jenkins

Тертычный Иван

Определимся с понятиями



Jenkins

Jenkins — проект для непрерывной интеграции с открытым исходным кодом, написанный на [Java](#)

Определимся с понятиями



Jenkins

Jenkins — проект для непрерывной интеграции с открытым исходным кодом, написанный на [Java](#)

Непрерывная интеграция (CI, [англ. *Continuous Integration*](#)) — это практика [разработки программного обеспечения](#), которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем.

- People
- Build History
- Project Relationship
- Check File Fingerprint
- Job Config History
- We Need Beer

Build Queue

No builds in the queue.

Build Executor Status

master

- 1 Idle
- 2 Idle

cabbage

- 1 Idle
- 2 Idle

celery

- 1 Idle
- 2 Idle

- All
- All Disabled
- All Failed
- All Unstable
- Infrastructure
- Jenkins core**
- Libraries

Other Projects

S	W	Name ↓	Last Success	Last Failure	Last Duration	LC
		jenkins_2.0	13 hr - #4	8 days 17 hr - #3	1 hr 2 min	
		jenkins_lts_branch	1 mo 5 days - #197	2 mo 9 days - #189	41 min	
		jenkins_main_maven-3.1.0	2 yr 5 mo - #7	N/A	1 hr 11 min	
		jenkins_main_trunk	2 days 5 hr - #4414	23 days - #4399	1 hr 1 min	
		jenkins_pom	1 yr 3 mo - #264	3 days 6 hr - #334	47 sec	
		jenkins_rc_branch	10 mo - #517	1 yr 7 mo - #424	25 min	
		remoting	2 yr 7 mo - #4	2 yr 7 mo - #3	6 min 19 sec	

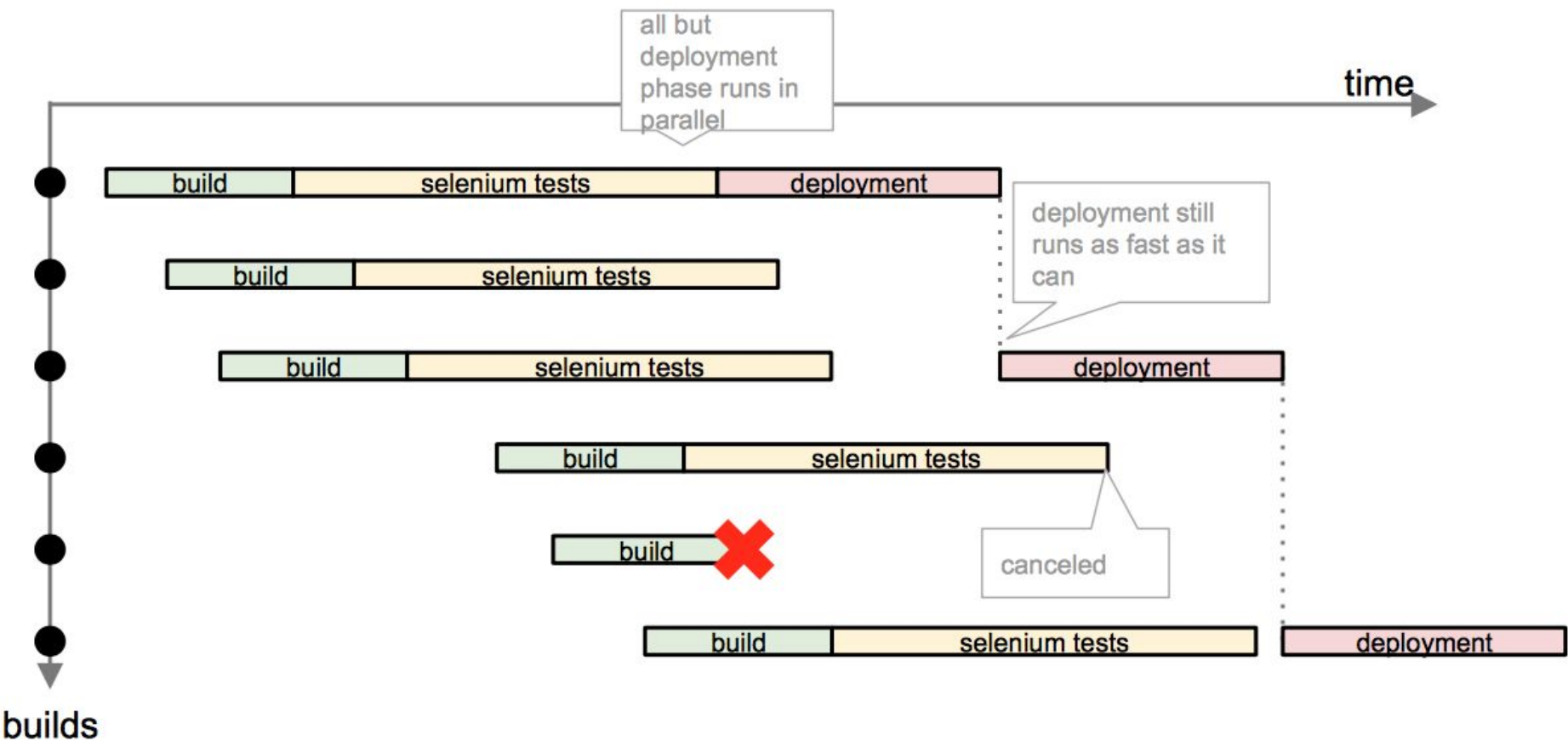
Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Project setup








- one workflow is defined as a job
- single script for all steps
- build triggers & parameters like regular projects
- SCM, publishing, &c. all part of script
- Each build shown using regular Jenkins view
- Graphical visualizations of actual build possible
 - (not of job definition; could be too dynamic)





Build

Триггеры сборки

- Trigger builds remotely (e.g., from scripts) 
- Build after other projects are built 
- Запускать периодически 
- Build when a change is pushed to BitBucket
- GitHub hook trigger for GITScm polling 
- Опрашивать SCM об изменениях 

Среда сборки

- Delete workspace before build starts
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Execute shell script on remote host using ssh 
- Use secret text(s) or file(s) 

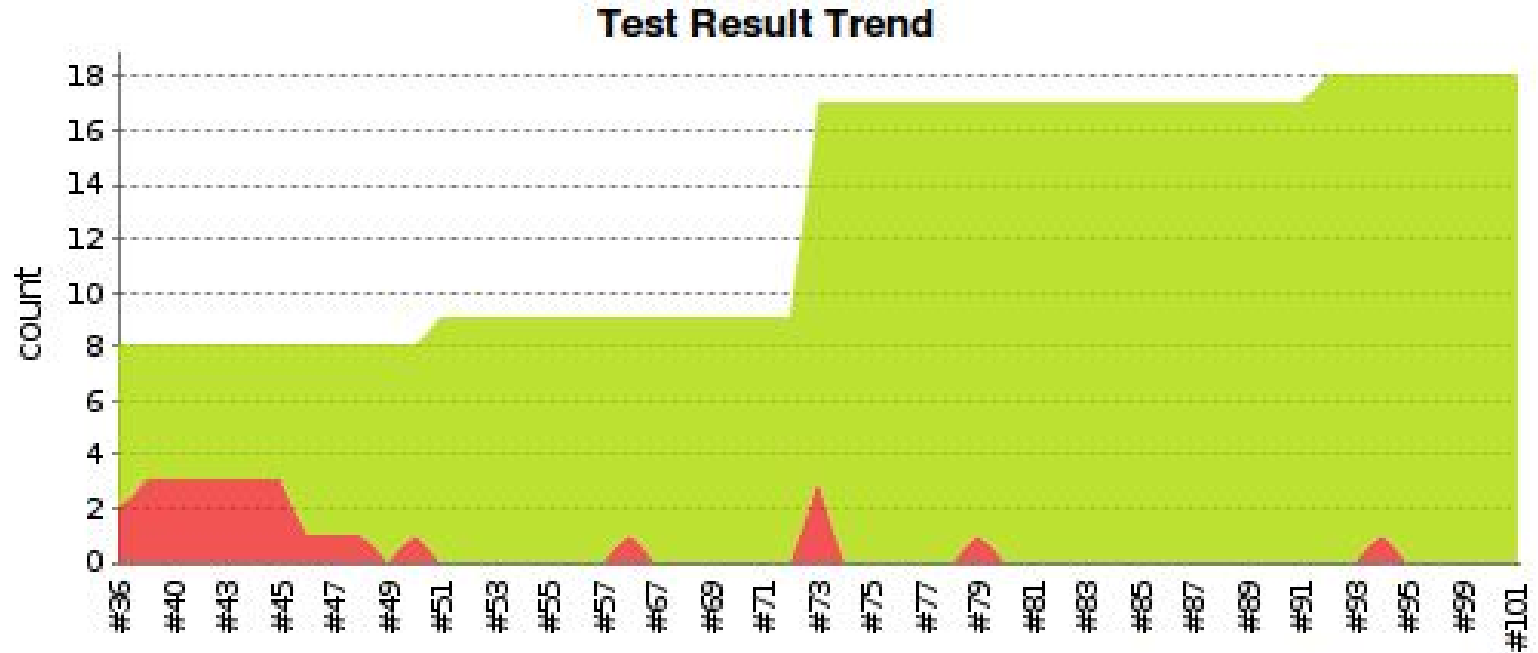
Сборка

Добавить шаг сборки ▾

- Execute shell script on remote host using ssh
- Invoke Gradle script
- Run with timeout
- Set build status to "pending" on GitHub commit
- Вызвать Ant
- Вызвать цели Maven верхнего уровня
- Выполнить команду Windows
- Выполнить команду shell

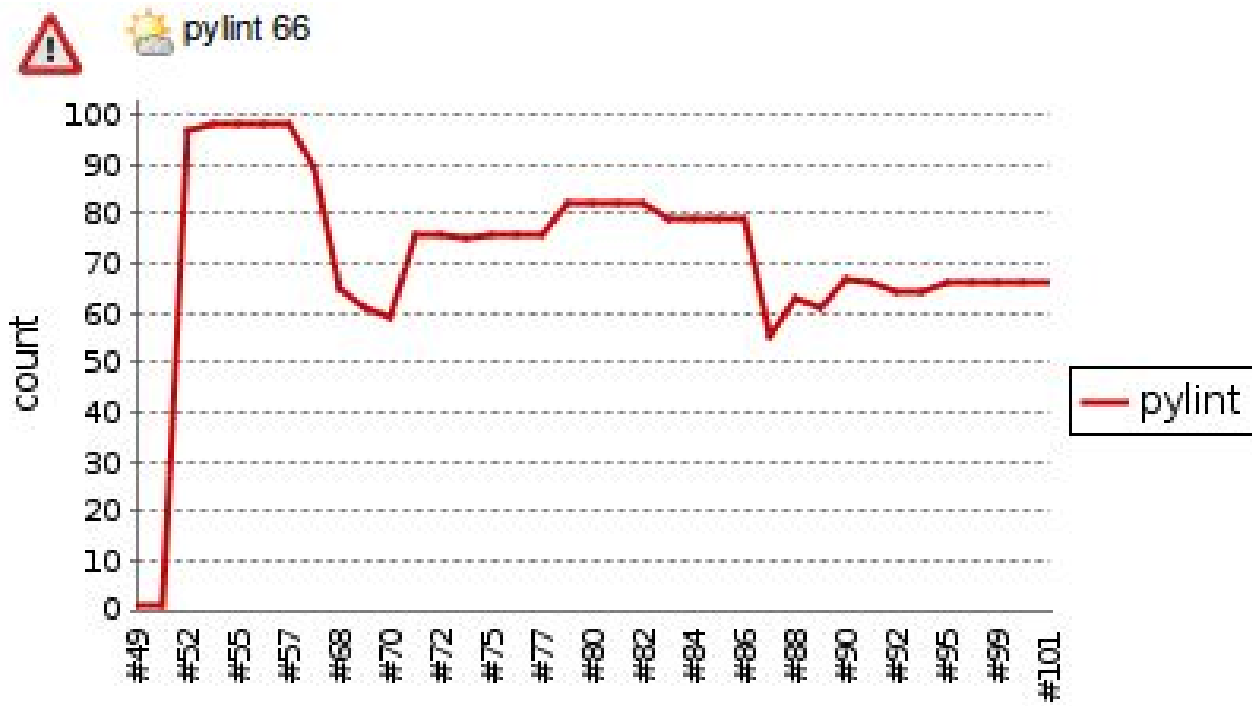
Test

nose2 + Cobertura



[\(just show failures\)](#) [enlarge](#)

pylint



Deploy

Name

Описание

Количество процессов-исполнителей

Корень удаленной ФС

Remote directory is mandatory

Метки

Использование

Способ запуска

Host

Credentials

The selected credentials cannot be found

Host Key Verification Strategy

Доступность

Node Properties

Environment variables

Список пар "ключ-значение"

Tool Locations



Jenkins

宛先: kyle

2014年2月22日 22:27

[詳細情報を隠す](#)

BuildAndTestHead - Build # 12 - Successful!

BuildAndTestHead - Build # 12 - Successful:

Check console output at <http://localhost:18080/job/BuildAndTestHead/12/> to view the results.



build.log

Визуализация





<https://habrahabr.ru/post/169097/>

Jenkins plays nicely with others



- Thanks to Jenkins' plugins and REST API, services and tools can easily interact with Jenkins and vice versa.
- Trigger builds based on GitHub pull requests, update JIRA upon successful builds and much, much more.
- I'll only touch on a few such tools and services - you can find many more on the Jenkins wiki.

Seven Habits of Highly Effective Jenkins Users (2014 edition!)

JIRA



- Update JIRA issues when commits with messages containing the issues are built.
- Follow build fingerprints to update issues in related projects as well.
- Generate JIRA release notes as part of the build process.

Seven Habits of Highly Effective Jenkins Users (2014 edition!)

Организация рабочего процесса

- 1) Maintain a single source repository
- 2) Automate the build
- 3) Make the build self-testing
- 4) Commit every day
- 5) Every commit builds on an integration machine

- 1) Keep the build fast
- 2) Test in a clone of the production environment
- 3) Make it easy to get the last executable
- 4) Everyone can see what's happening
- 5) Automate deployment

Рабочий процесс в GitHub

- Содержимое ветви `master` всегда работоспособно (*deployable*).
- Начиная работу над чем-то новым, ответвляйте от ветви `master` новую ветвь, имя которой соответствует её предназначению (например, «`new-oauth2-scopes`»).
- Закоммитив в эту ветвь локально, отправляйте регулярно свою работу и в одноимённую ветвь на сервере.
- Когда вам понадобится отзыв, или помощь, или когда вы сочтёте ветвь готовою ко слиянию, отправьте **запрос на слияние**.
- После того, как кто-то другой просмотрел и одобрил фичу, вы можете слить вашу ветвь в ветвь `master`.
- После того, как ветвь `master` пополнилась новым кодом, вы можете немедленно внедрить его на продакшен и вам следует сделать это.

Спасибо за внимание!