Jenkins, votre serviteur

C. Loomis (CNRS/LAL)

Journée LoOPS

11 décembre 2012

Intégration Continue : Wikipedia

L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée.

Intégration Continue : Pratique

- Etre plus confiant que un logiciel fonctionne correctement et efficacement
  - Améliorer la qualité du code
  - Vérifier le comportement du code avec les tests unitaires
  - Déployer un système complet et valider la fonctionnalités

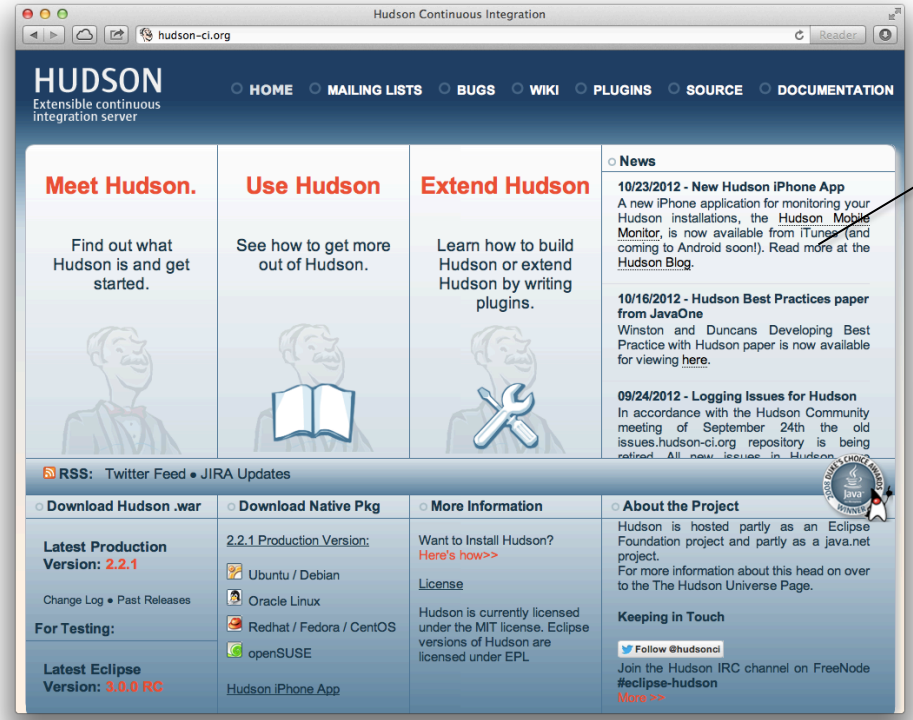
- Tous pour chaque changement du code!

Les Outils Automatisés

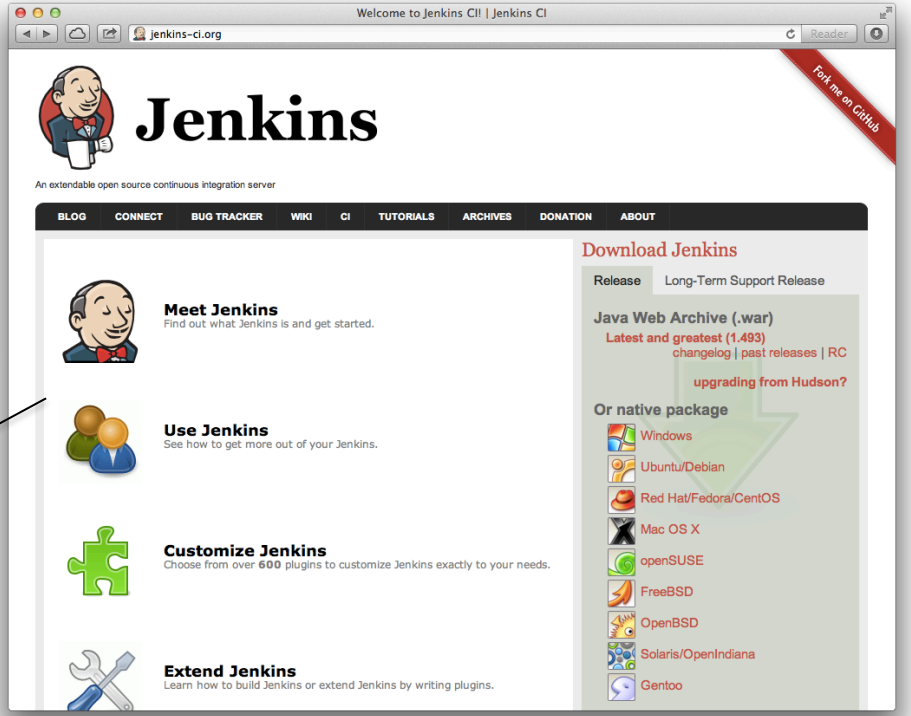
- L'intégration continue est possible avec un minimum d'effort humain grâce à des outils automatisés

- Open Source
  - Hudson, Jenkins, Travis-CI, CruiseControl, …
- Commerciaux
  - Bamboo, TeamCity, …

Hudson & Jenkins



Hudson
http://hudson-ci.org



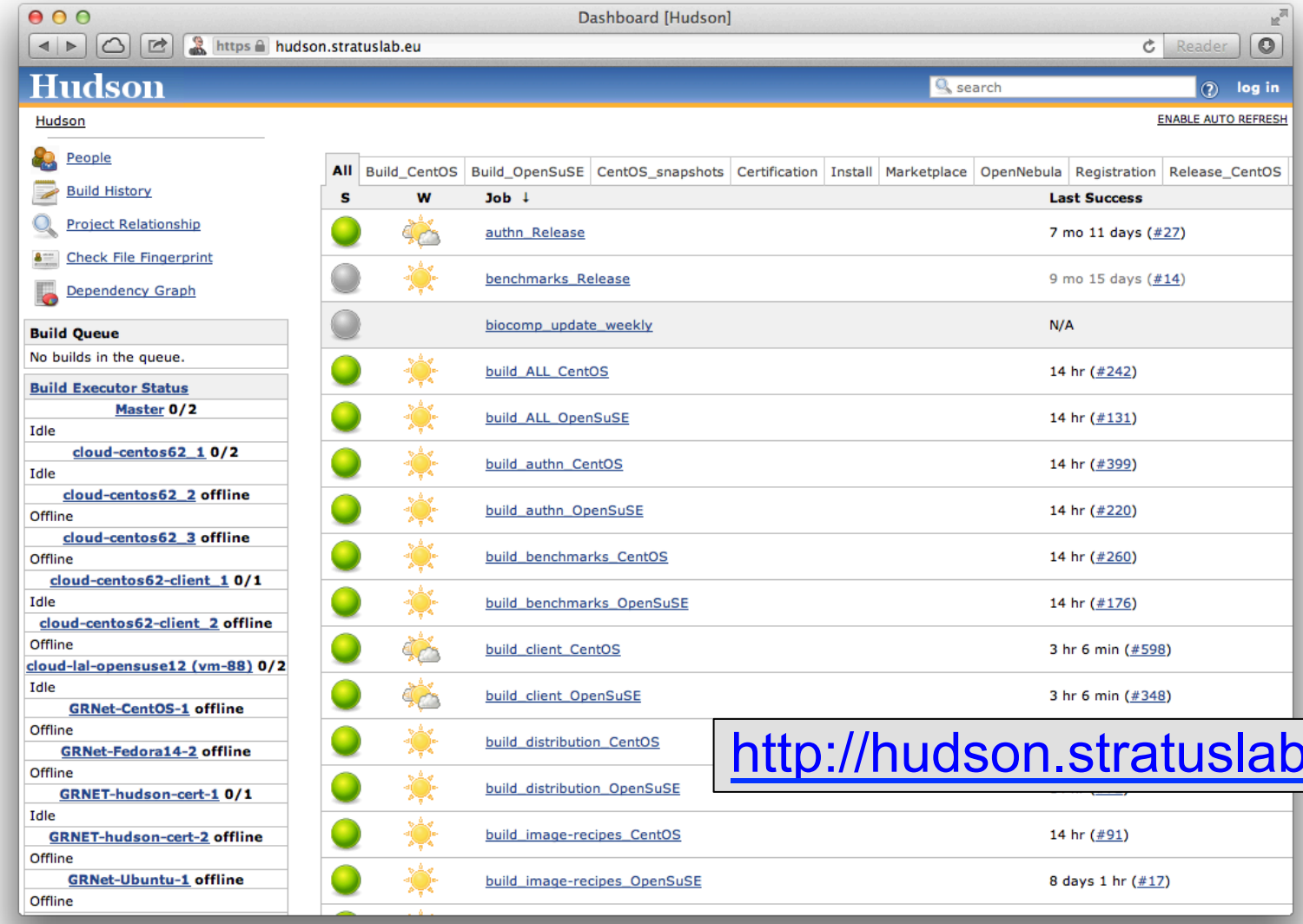
Jenkins
http://jenkins-ci.org

Le « Fork » Hudson/Jenkins

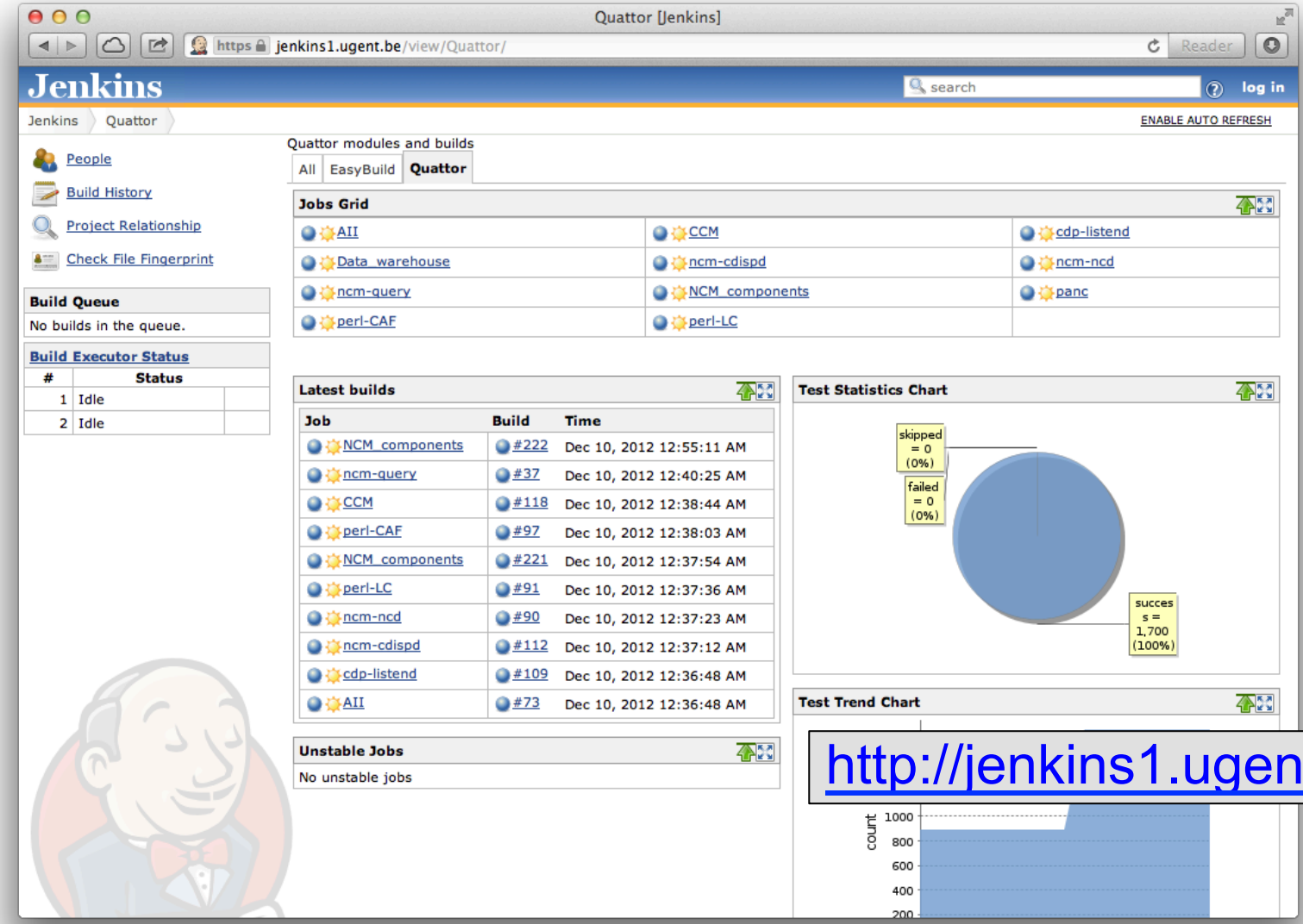
- L'existence des deux produits quasi-identiques est un (des plusieurs) effet de bord de l'acquisition du Sun par Oracle
  - La communauté open source (responsable pour une grand nombre de plugins) a crée Jenkins
  - Oracle/Sonatype continue la développement du Hudson, mais dans le Eclipse Foundation

- Malheureusement trop tard pour fusionner les communautés au tour d'une produit unique

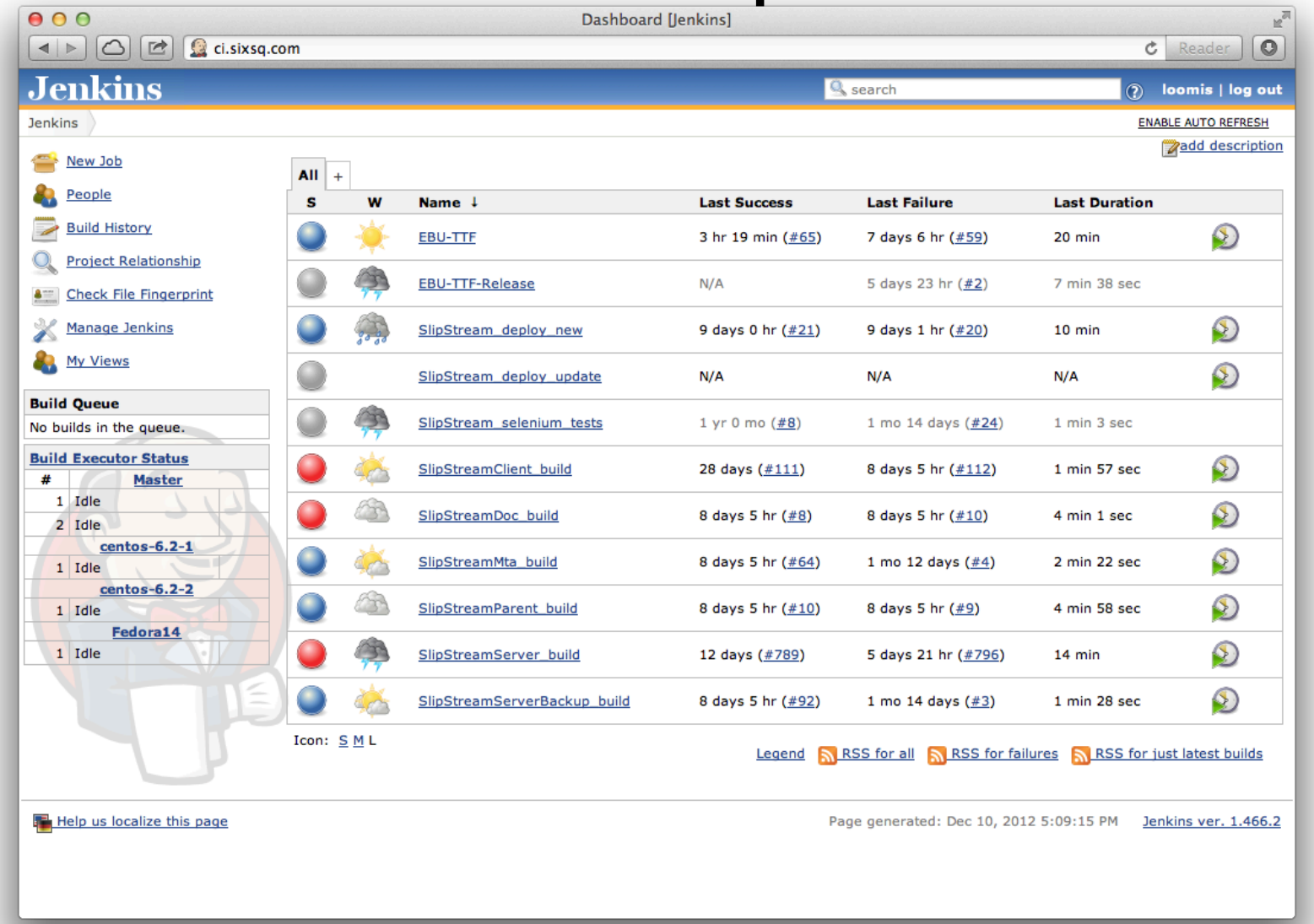
StratusLab


Dashboard [Hudson]
https hudson.stratuslab.eu
Reader
Hudson
search
log in
Hudson
ENABLE AUTO REFRESH
People
Build History
Project Relationship
Check File Fingerprint
Dependency Graph
Build Queue
No builds in the queue.
Build Executor Status
Master 0/2
Idle
cloud-centos62_1 0/2
Idle
cloud-centos62_2 offline
Offline
cloud-centos62_3 offline
Offline
cloud-centos62-client_1 0/1
Idle
cloud-centos62-client_2 offline
Offline
cloud-lal-opensuse12 (vm-88) 0/2
Idle
GRNet-CentOS-1 offline
Offline
GRNet-Fedora14-2 offline
Offline
GRNET-hudson-cert-1 0/1
Idle
GRNET-hudson-cert-2 offline
Offline
GRNet-Ubuntu-1 offline
Offline
All
Build_CentOS
Build_OpenSuSE
CentOS_snapshots
Certification
Install
Marketplace
OpenNebula
Registration
Release_CentOS
S
W
Job
Last Success
authn_Release
7 mo 11 days (#27)
benchmarks_Release
9 mo 15 days (#14)
biocomp_update_weekly
N/A
build_ALL_CentOS
14 hr (#242)
build_ALL_OpenSuSE
14 hr (#131)
build_authn_CentOS
14 hr (#399)
build_authn_OpenSuSE
14 hr (#220)
build_benchmarks_CentOS
14 hr (#260)
build_benchmarks_OpenSuSE
14 hr (#176)
build_client_CentOS
3 hr 6 min (#598)
build_client_OpenSuSE
3 hr 6 min (#348)
build_distribution_CentOS
build_distribution_OpenSuSE
build_image-recipes_CentOS
14 hr (#91)
build_image-recipes_OpenSuSE
8 days 1 hr (#17)
http://hudson.stratuslab.eu/

Quattor


Quattor [Jenkins]
https jenkins1.ugent.be/view/Quattor/
Reader
Jenkins
search
log in
Jenkins
Quattor
ENABLE AUTO REFRESH
People
Build History
Project Relationship
Check File Fingerprint
Build Queue
No builds in the queue.
Build Executor Status
Status
1 Idle
2 Idle
Quattor modules and builds
All
EasyBuild
Quattor
Jobs Grid
AII
CCM
cdp-listend
Data_warehouse
ncm-cdispd
ncm-ncd
ncm-query
NCM_components
panc
perl-CAF
perl-LC
Latest builds
Job Build Time
NCM_components #222 Dec 10, 2012 12:55:11 AM
ncm-query #37 Dec 10, 2012 12:40:25 AM
CCM #118 Dec 10, 2012 12:38:44 AM
perl-CAF #97 Dec 10, 2012 12:38:03 AM
NCM_components #221 Dec 10, 2012 12:37:54 AM
perl-LC #91 Dec 10, 2012 12:37:36 AM
ncm-ncd #90 Dec 10, 2012 12:37:23 AM
ncm-cdispd #112 Dec 10, 2012 12:37:12 AM
cdp-listend #109 Dec 10, 2012 12:36:48 AM
AII #73 Dec 10, 2012 12:36:48 AM
Test Statistics Chart
skipped = 0 (0%)
failed = 0 (0%)
succes s = 1,700 (100%)
Test Trend Chart
count
1000
800
600
400
200
Unstable Jobs
No unstable jobs
http://jenkins1.ugent.be/

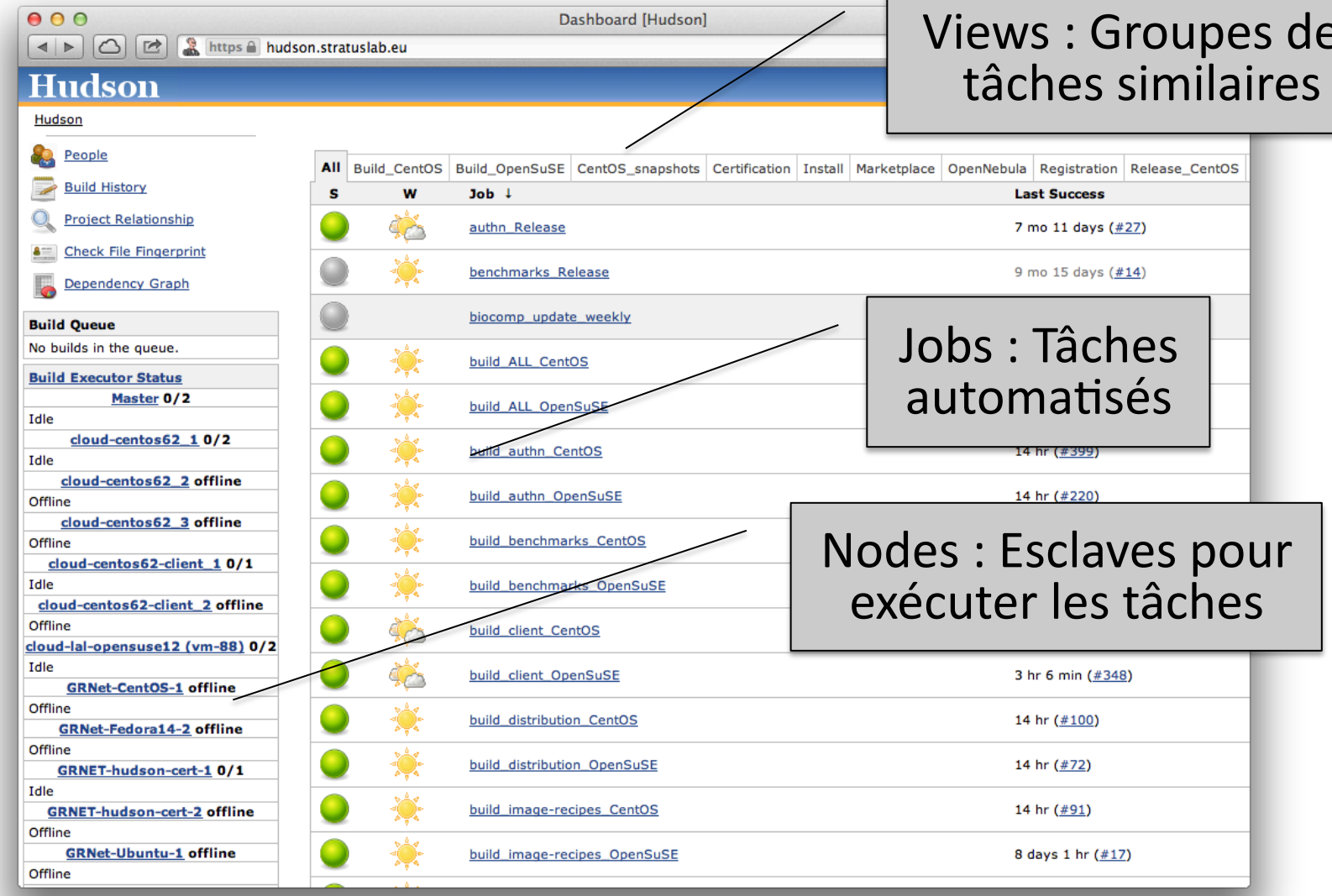
SixSq


Dashboard [Jenkins]
ci.sixsq.com
Reader
Jenkins
search
loomis | log out
Jenkins
ENABLE AUTO REFRESH
add description
New Job
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
Master
1 Idle
2 Idle
centos-6.2-1
1 Idle
centos-6.2-2
1 Idle
Fedora14
1 Idle
All
S W Name ↓ Last Success Last Failure Last Duration
EBU-TTF 3 hr 19 min (#65) 7 days 6 hr (#59) 20 min
EBU-TTF-Release N/A 5 days 23 hr (#2) 7 min 38 sec
SlipStream_deploy_new 9 days 0 hr (#21) 9 days 1 hr (#20) 10 min
SlipStream_deploy_update N/A N/A N/A
SlipStream_selenium_tests 1 yr 0 mo (#8) 1 mo 14 days (#24) 1 min 3 sec
SlipStreamClient_build 28 days (#111) 8 days 5 hr (#112) 1 min 57 sec
SlipStreamDoc_build 8 days 5 hr (#8) 8 days 5 hr (#10) 4 min 1 sec
SlipStreamMta_build 8 days 5 hr (#64) 1 mo 12 days (#4) 2 min 22 sec
SlipStreamParent_build 8 days 5 hr (#10) 8 days 5 hr (#9) 4 min 58 sec
SlipStreamServer_build 12 days (#789) 5 days 21 hr (#796) 14 min
SlipStreamServerBackup_build 8 days 5 hr (#92) 1 mo 14 days (#3) 1 min 28 sec
Icon: S M L
Legend RSS for all RSS for failures RSS for just latest builds
Help us localize this page
Page generated: Dec 10, 2012 5:09:15 PM
Jenkins ver. 1.466.2

StratusLab

- Projet européen et maintenant communauté ouverte pour la développement d'une un logiciel cloud « IaaS »
- Utiliser Hudson pour 2.5+ ans pour automatiser les tâches de développement
  - Testing Continu
  - Intégration Continue
  - Gestion des Releases

Vocabulaire

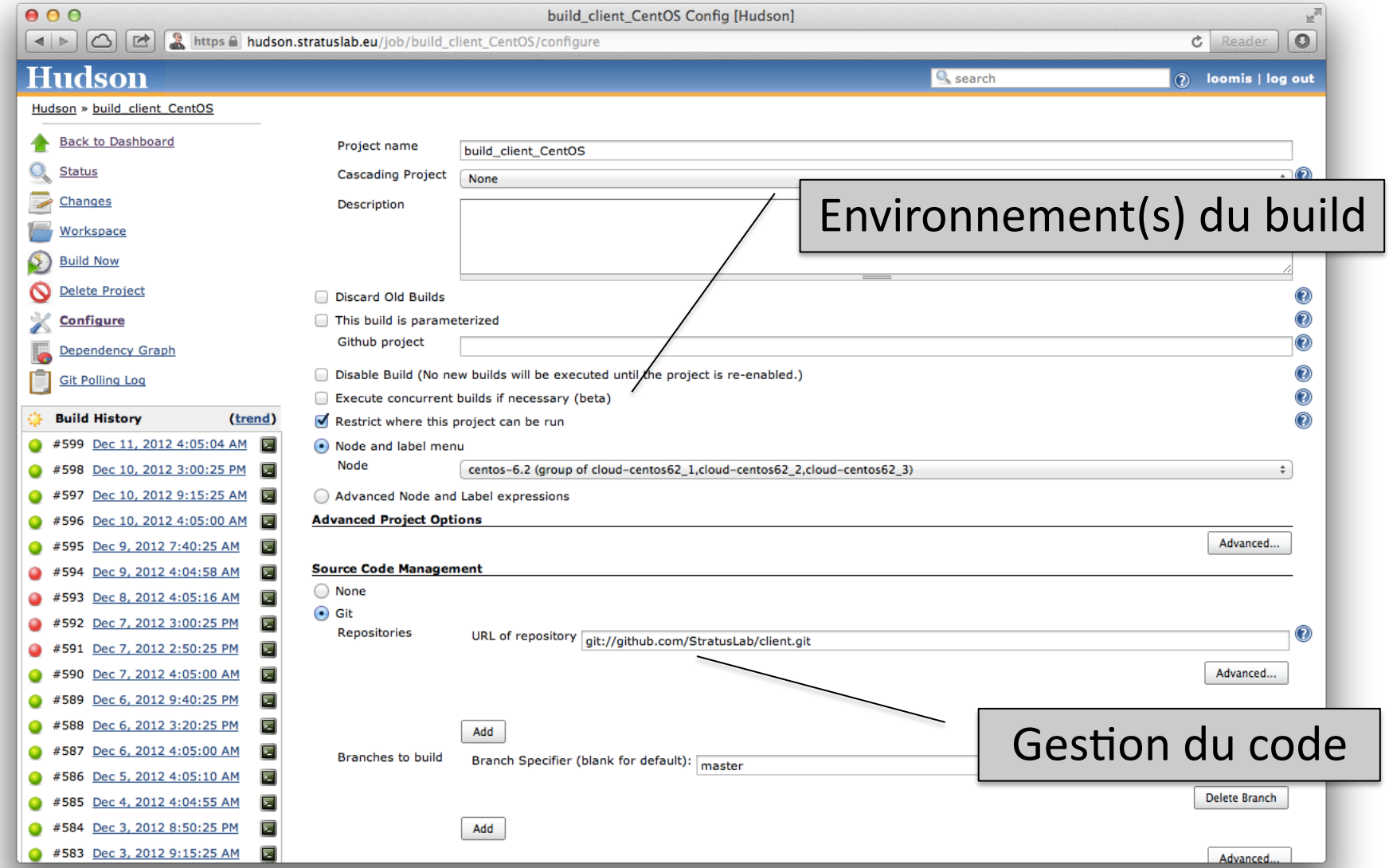


Views : Groupes des tâches similaires

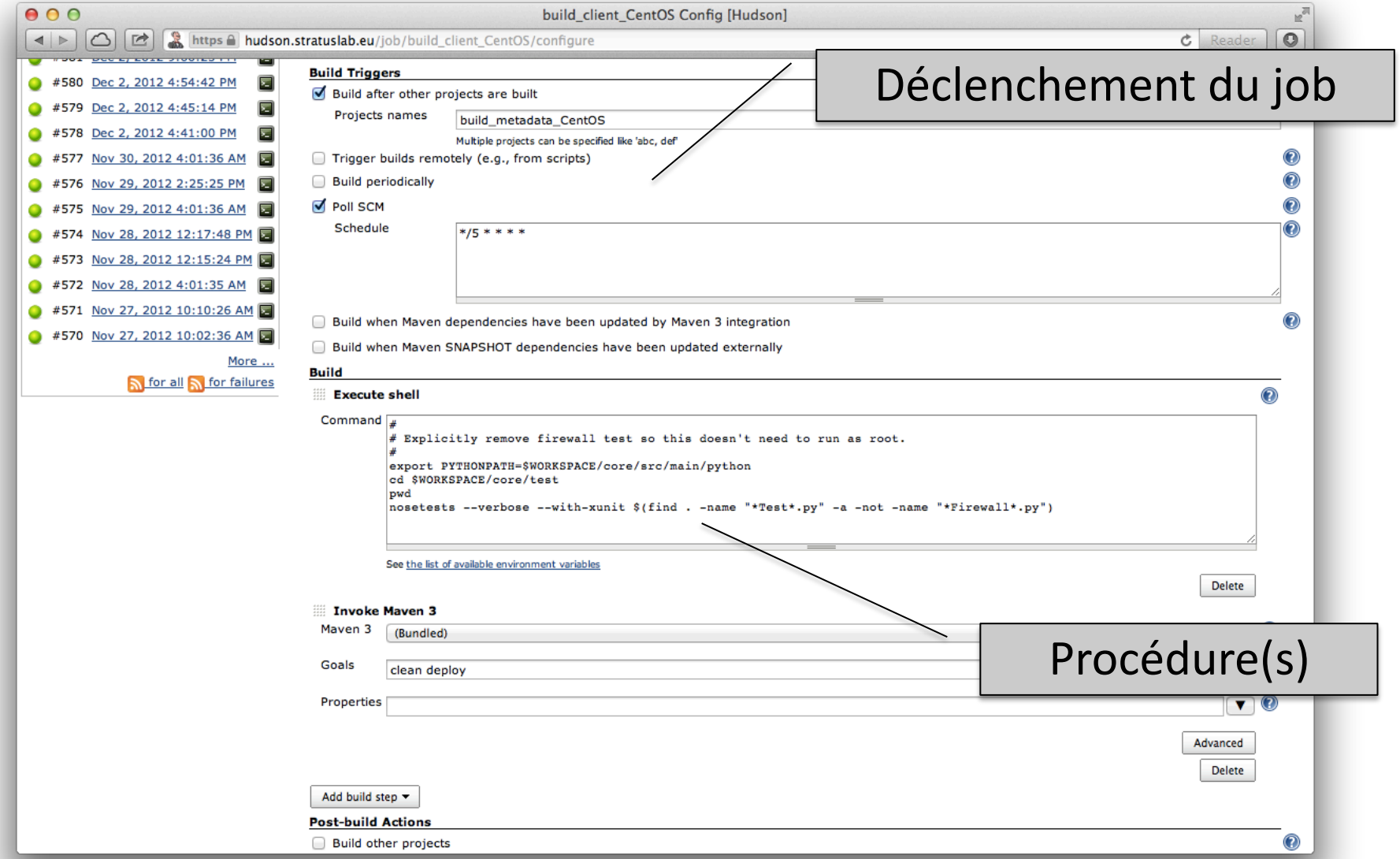
Jobs : Tâches automatisés

Nodes : Esclaves pour exécuter les tâches

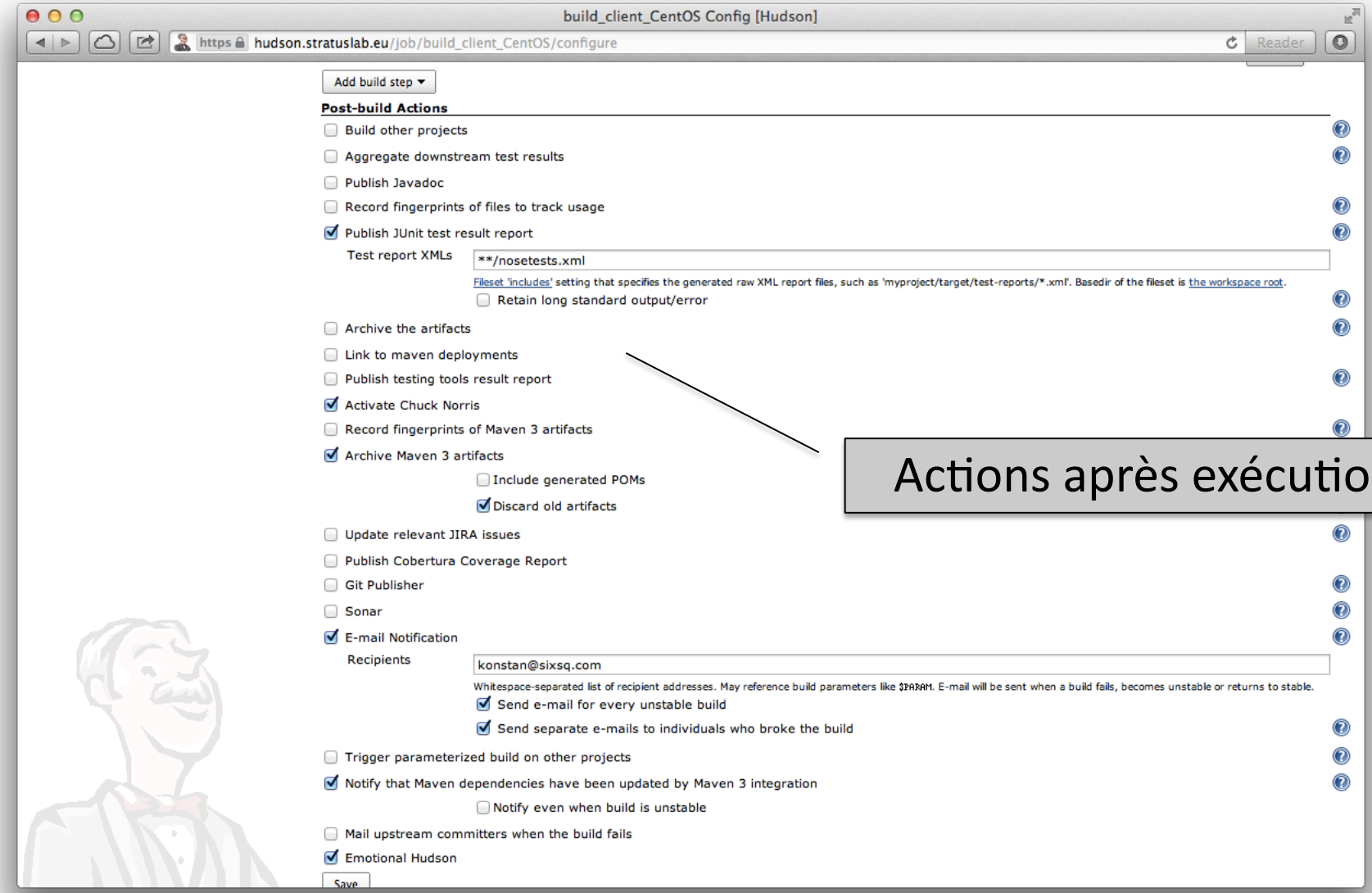
Définition d'un Job



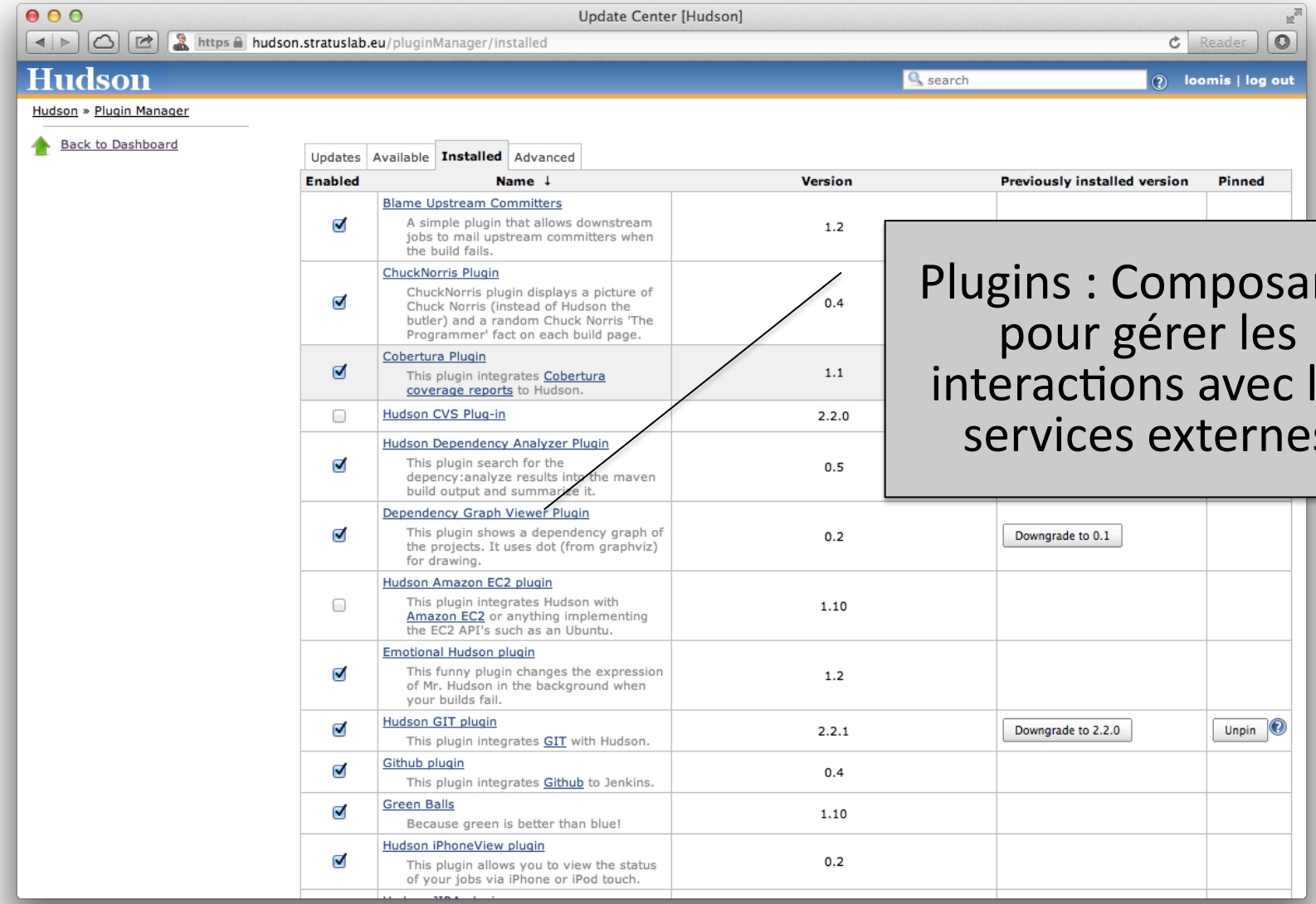
Définition d'un Job



Définition d'un Job

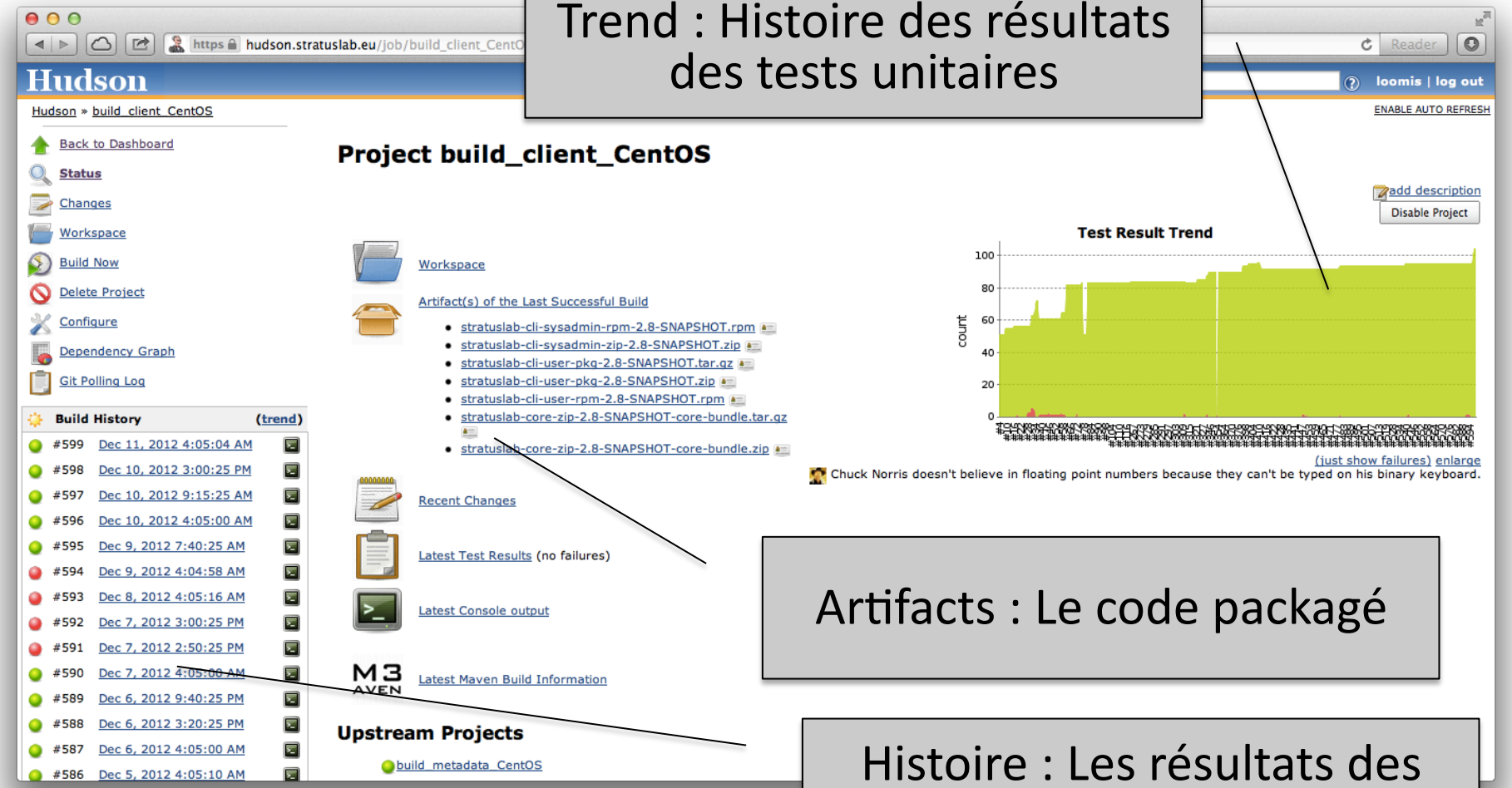


Vocabulaire



Plugins : Composants pour gérer les interactions avec les services externes

Vocabulaire



Testing Continu

- Exécuter les tests unitaires sur plusieurs OS après chaque commit
  - Bon intégration avec les gestionnaires du code : git, svn, cvs, …
  - Définir une matrice des environnement différentes : OS, java, …
  - Les jobs peuvent être déclencher automatiquement avec notification des résultats
  - Supporte excellente pour maven

Contrôle de Qualité

- Pour StratusLab est intégré avec les tests unitaires et automatiser au maximum
- FindBugs – analyseur statique pour Java
  - Traiter comme un test unitaire pour les composants StratusLab écrit en Java
  - Ça trouve les problèmes de codage fréquents et forme les développeurs non-experts en Java
  - Les « code reviews » peuvent concentre sur les problèmes de design et architecture

Travis-CI

- Après son passage à GitHub, StratusLab utilise aussi Travis-CI pour le testing continu
  - + Les tests sont déclencher automatiquement avec les commits dans GitHub
  - + On peut afficher un badge « pass, fail » dans le README du projet
  - + Rien a déployer : les esclaves sont déjà là
  - – Les environnements disponibles sont assez limité
  - – Des tests avec un faux résulte sont fréquents

Intégration Continue

- Créer toutes les composants nécessaires pour un système complet
  - Construire les packages (CentOS, OpenSuSE)
  - Déployer un cloud test avec ces packages
  - Faire des tests fonctionnels dans le cloud test
- Jenkins peut gérer les dépendances entre les jobs quasi-automatiquement
- Peut stocker les artifacts dans Jenkins

Dépendances StratusLab

Gestion de Releases

- Faire toutes les actions pour un release final
  - Le tagging du code est faites par les jobs hudson
  - Génération des packages finals
  - Création des dépôts YUM
  - Installation et vérification du système dans une infrastructure pour la certification

Points Forts

- Jenkins/Hudson est stable et faible
- Simple de déployer et configurer
- Milliers des plugins pour automatiser les tâches
- Très bonne intégration avec les outils du build
- Très bonne intégration avec les outils du test
- Histoire du statut du job et une résumé des changements du code
- Facile de ajouter les nouveau jobs et éditer les jobs existants

Pas Complètement Automatique

- Ça prendre du temps pour maintenir les jobs dans Jenkins et ajouter des nouveaux
- Doit avoir quelqu'un(e) qui surveiller les résultats et forcer les gens de corriger leurs bugs
- Ça prend du temps pour créer et maintenir les machines avec le(s) bonne(s) environnement(s) pour le build et test
- L'intégration avec les ressources cloud n'est pas très utile a ce moment

Points Négatifs

- Un serveur Jenkins pour chaque projet
  - Le contrôle d'accès et le gesion des views sont très limité
- Les configuration peut venir très complexe
- On peut facilement cacher des informations utiles dans les jobs Jenkins : p.e. les astuces pour la déploiement
- Les configurations du jobs ne sont pas versionner : difficile de distinguer si un job échoue au cause d'une changement du code ou du test
- Les tests du système prenant beaucoup de temps : les changement sont mélangés et le feedback au développeurs est retardé

Bénéfices

- Tableau de bord pour comprendre le statut du logiciel
- Minimisation du effort pour mettre en place les processus de développement et les maintenir
- Ça rendre accessible les informations du build, test, et intégration au toutes les participants

Résumé

- Tous les projets de développement doivent utiliser un système de l'intégration continue
  - Ça valider les changement du code rapidement
  - Permet la validation dans plusieurs environnements
  - Faciliter les discussions des problèmes/changements
- Jenkins/Hudson
  - Bon candidat pour vos projets : facile à déployer mais robuste et complet
  - Suffisamment flexible pour automatiser les test unitaires, contrôles de qualité, l'intégration des composants, et le gestion des releases

Questions?