



Intégration continue

Intégration quoi ?

L'intégration continue est un ensemble de pratiques utilisées en génie logiciel. Elles consistent à **vérifier automatiquement à chaque modification de code source que le résultat des modifications ne produit pas de régression** de l'application en cours de développement.

— [Wikipedia](#)

En français ?

T'as le droit de tout péter, mais t'es obligé de **mettre les autres au courant**.

— @n1k0

Le but de l'intégration continue

- Avoir **confiance** en son propre code et celui de l'équipe
- Renforcer la qualité du code en **responsabilisant** l'équipe
- Déployer plus **sereinement**
- Soulager les perfectionnistes pressés et sensibles au **stress**
- Prendre l'**apéro** plus tôt

Comment ?

Régulièrement et automatiquement :

- Contrôler l'**intégrité** unitaire et fonctionnelle
- S'assurer de l'absence de **régressions**
- Surveiller le respect des **conventions** de codage
- Mesurer la **couverture** du code
- Signaler les **erreurs** de codage
- Générer des **rapports** synthétiques utiles

Présentations

- Laurent/Cal : Jenkins
- Christian A. : l'approche Nuxeo
- Sébastien : Travis-CI
- Oleg : Sonar
- Éric Legay (CSNSM) : une ressource IN2P3 ?

Jenkins

Deploiement et Utilisation

- StratusLab (<https://hudson.stratuslab.eu/>)
 - « Continuous Testing »: Exécuter les tests unitaires sur plusieurs OS après chaque commit
 - « Continuous Integration »: Construire les packages, déployer un système et faire les tests fonctionnels
 - « Release Management »: Générer les packages, certifier le release, et créer les dépôts YUM
- PSPA (134.158.75.115:8080)
 - Tests continus : Compilation et exécution sur une machine cible SL6 après chaque commit
 - Déploiement d'une application web

Jenkins

Jenkins

•PSPA



The screenshot shows the Jenkins web interface. At the top, there is a search bar with the text "rechercher" and a "Jenkins" logo. Below the search bar, there are navigation links: "Nouveau Job", "Utilisateurs", "Historique des constructions", and "Administrer Jenkins". On the right side, there are links for "ENABLE AUTO" and "ajouter une d".

The main content area displays a table of build jobs. The table has columns for "S" (Success), "W" (Warning), "Name", "Dernier Succès", "Dernier Echec", and "Dernière Durée". There are three rows of jobs listed:

S	W	Name	Dernier Succès	Dernier Echec	Dernière Durée
		Geant	7 j 5 h (#30)	1 j 1 h (#24)	1 mn 38 s
		PSPA-Dev	5 j 5 h (#73)	4 mn 42 s (#106)	1 mn 57 s
		PSPA-stable	N/A	1 h 17 mn (#31)	15 mn

Below the table, there is a legend for icons: "Icône: S M L". At the bottom right, there are links for "Légende", "RSS pour toutes", "RSS pour tous les échecs", and "RSS juste pour les derniers".

On the left side, there are sections for "File d'attente des constructions" (empty) and "État du lanceur de constructions" with a table:

#	Statut
1	En attente
2	En attente

Jenkins

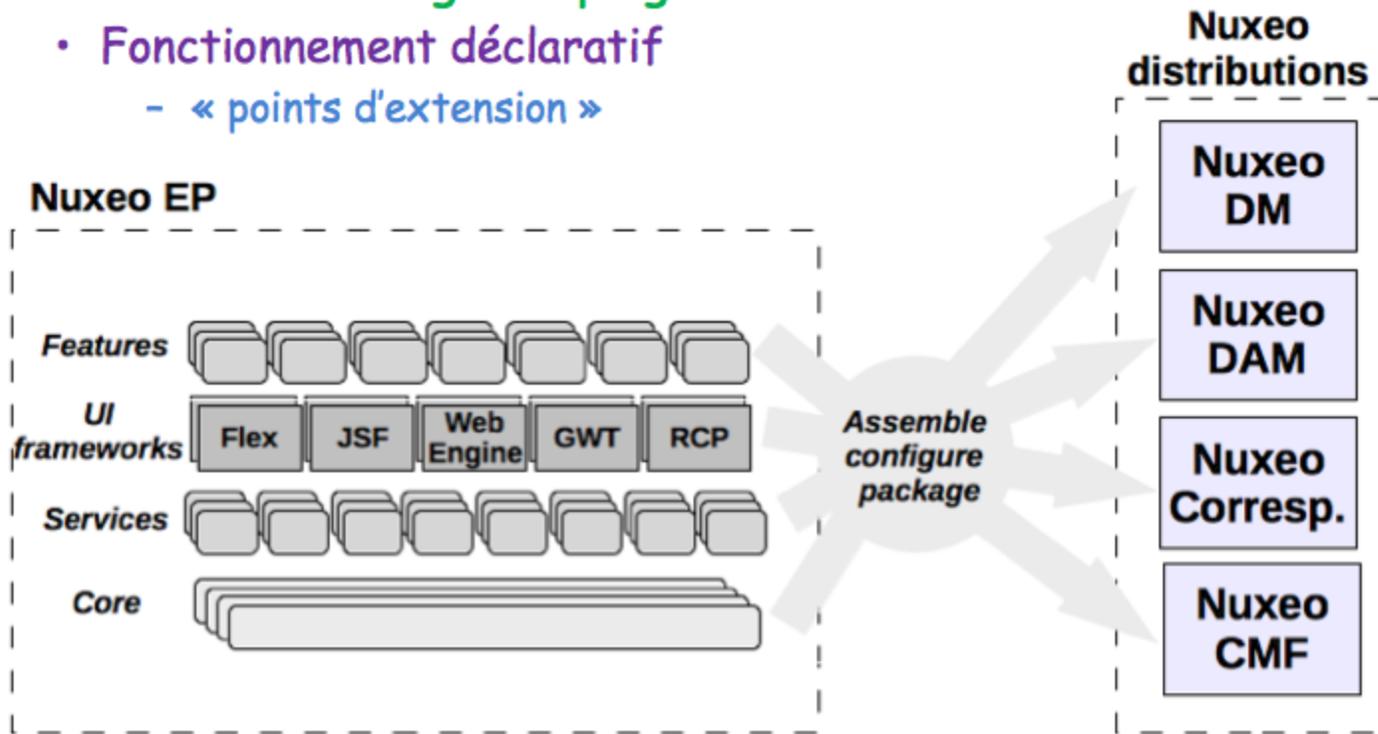
Avantages et Inconvénients

- Utilisation simple vers une interface web
 - Bonne intégration avec les SCMs
 - Gestion des plusieurs OS simultanément
 - Dépendances possible entre tâches
- Pas d'historique des configuration de test
 - Les configurations ne sont pas sauvegardées

Approche Nuxeo

Développer avec Nuxeo

- Le logiciel Nuxeo
 - Open source
 - Architecture client-server
 - C'est un assemblage de plugins Java/XML
 - Fonctionnement déclaratif
 - « points d'extension »



Approche Nuxeo

Développer avec Nuxeo

- Eclipse
 - Nuxeo a fourni un plugin « nuxeo »
 - Plus un Wizard pour construire des Nuxeo Artifact
 - Operation
 - Component
 - Document adapter
 - Document Listener
 - Automation gadget
 - Action/Controller/Service Bean
 - Security
 - Lancement et debugging du server à travers Eclipse
 - Déploiement des artifacts développés
 - Construction d'un plugin Nuxeo (construction, configuration, packaging)
 - Un plugin Nuxeo peut être déployé à partir de l'Admin Center du client Nuxeo (donc sans visibilité directe du code du serveur)

Approche Nuxeo

Développer avec Nuxeo

- Maven
 - Est un autre moyen de configurer/packager/déployer des plugin nuxeo
 - L'environnement Eclipse sait configurer le *pom.xml* correspondant
- Junit
 - Le plugin Nuxeo pour Eclipse fournit aussi un plugin Junit spécialisé pour Nuxeo.
 - Templates de tests pour chaque type d'artefact
 - Possibilité de configurer un environnement Jenkins pour automatiser les tests ainsi définis
- Nuxeo Shell
 - Client mode line qui permet de naviguer dans la base des plugins installés
 - Permet d'activer manuellement les opérations, les tests
 - Scripting pour lancer des batches

Travis-CI

A hosted continuous integration service for the open source community.

- <https://travis-ci.org/>
 - MIT license
 - written in Ruby (nobody's perfect)
 - leverages vagrant for VM provisioning
 - **integrated** w/ GitHub
 - first class support for:
 - ▶ C, C++, Clojure, Erlang, **Go**, Groovy, Haskell,
 - ▶ Java, JavaScript (with Node.js), Perl, PHP, Python, Ruby, Scala
 - VM: ubuntu-32b-12.04
 - configuration thru a **YAML** file: `.travis.yml`
 - a **Go** example: `go-ffi`
-

Travis-CI

```
language: go
```

```
before_install:
```

- sudo apt-get update -qq
- sudo apt-get install libffi-dev -qq

```
install: go get -d -v ./pkg/... && go build -v ./pkg/...
```

```
script: go test ./pkg/...
```

```
notifications:
```

```
  email:
```

```
    recipients:
```

- binet@cern.ch

```
    on_success: change
```

```
    on_failure: always
```

Sonar

Objectifs

- ❑ améliorer la lisibilité du code
- ❑ se conformer aux standards d'écriture
- ❑ vérifier le niveau de documentation in-ligne

Dissémination

Debugging

- ❑ détecter les duplications de code
- ❑ implémenter les tests unitaires
- ❑ contrôler le taux de couverture
- ❑ détecter les codes «morts»

- ❑ et toujours : le gestion des versions

Versionning

Sonar

Chaîne d'outils

Trop (de docs, d'outils...) peut être aussi néfaste que pas assez

Dissémination

- Eclipse (<http://www.eclipse.org>)
- Sonar (<http://www.sonarsource.org>)

Debugging

- JUnit (<http://junit.sourceforge.net>)

versionning

- svn (<http://subversion.tigris.org>)

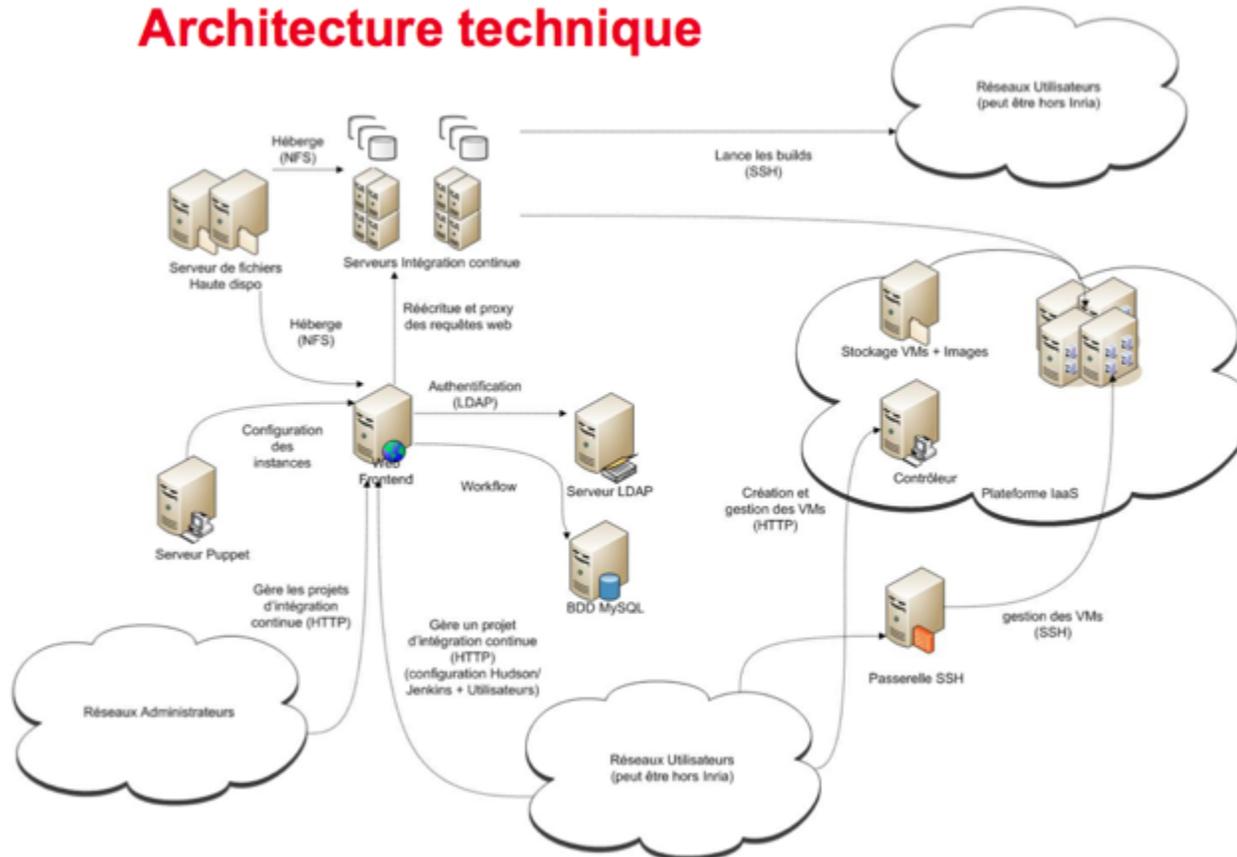
Une ressource IN2P3 ?



**Plateforme nationale
d'intégration continue**

Une ressource IN2P3 ?

Architecture technique



2e journée *LoOPS*

- LoOPS: développements logiciels Pour la Science (ou Paris Saclay)
- 11 décembre dans le "Grand amphi" du bâtiment 640 de l'Université Paris-Sud, sur le plateau de Saclay.

Cette journée sera axée sur le test et le build. Elle mêlera des retours d'expériences, des ateliers et des présentations de pratiques et d'outils.

	9h00	Accueil	
Exposés pléniers	9h15	Les tests, comment les aborder ? (Frédéric Voisin, équipe ForTesSE, LRI)	
	10h00	Retour d'expérience avec Sonar (Oleg Lodygensky, LAL)	
	10h30	Pause	
	11h00	Jenkins, votre serviteur (Charles Loomis, LAL)	
	11h45	N'ayez pas peur d'OCaml (Maxence Guesdon, INRIA Saclay)	
	12h00	Utiliser Cmake pour la reconstruction, les tests, le packaging (Alexandre Abadie, INRIA Saclay)	
	12h20	Déjeuner	
Ateliers parallèles	13h30	Atelier Cmake	Atelier OCaml
	15h45	Pause	
	16h00	Table ronde: retours sur la journée, questions, JDEV2013, suite du réseau	
	17h00	Fin de la journée	

Voilà !