



# Workshop ESUP4

Paris, 12 Mai 2014

# Plan - (prévision)

## Matin

Présentation Git

EsupV4

- ✓ Recommandations (ou parti pris)
- ✓ Présentation de la VM
- ✓ Installation/Configuration basique EsupPortail

## Après-Midi

Look / ergonomie

Intégration Iframe

Merge et cherry-pick

Single LogOut



# Présentation Git

Paris, 12 Mai 2014

# Sommaire

**Outils de gestion de version**

**GIT – outil de versioning**

**GIT – en quelques lignes de commandes**

**GIT – fonctionnement**

**GIT – intérêt pour l'exploitant**

# versioning dans la vie courante

**Pour gérer l'écriture d'un article, d'une thèse ...**

**Outils bureautique :**

Diff Word

Record Modifications ...

**GED**

**Wiki ...**

# Sans outil de gestion de versions ?

On fait des copies de fichiers de conf en .back, .ori,  
.sav\_200113, \_init

On fait des tar gz

On s'envoie ces sauvegardes par mail, on les stocke sur son  
poste, un disque, une clef ...

-> on fait du versioning à la main !

C'est déjà pas si mal : mieux que ne pas versionner du tout.

Mais des outils plus haut niveau de gestion de versions  
permettent de gagner en efficacité ... et plus encore :)

# Outils de gestion de version

## Historique

SCCS – 1972

RCS – 1982

CVS – 1986

SVN – 2000

# GIT

## GIT – 2005

Logiciel de gestion de versions décentralisée  
Créé par Linus Torvald pour le dev du noyau  
linux

## Github – 2008

~ réseau social autour de Git

\* **Confère wikipedia** : Le magazine PC World nous apprend que « quand on lui a demandé pourquoi il avait appelé son logiciel "git", qui est à peu près l'équivalent de "connard" en argot britannique<sup>45</sup>, Linus Torvalds a répondu "je ne suis qu'un sale égoцентриque, donc j'appelle tous mes projets d'après ma propre personne. D'abord Linux, puis Git." ».



# GIT comme outil du développeur

## **Pour 1 développeur :**

historiser les itérations du dev ~ sauvegardes

## **Pour N développeurs :**

travailler ensemble en même temps sur 1 même code (partage de code, gestion de conflits)

## **Pour 1 chef de projet :**

suivre les devs de chacun, intégration des contributions ...

# GIT – commandes de base

**Sous linux, apt-get, yum & co suffisent pour l'installation.**

apt-get install git-core

**... et dans un répertoire avec des fichiers de config ...**

git init                   -> entrepôt git créé

git add \*                 -> fichiers prêts à être effectivement versionnés

git commit -m «configs initiales »

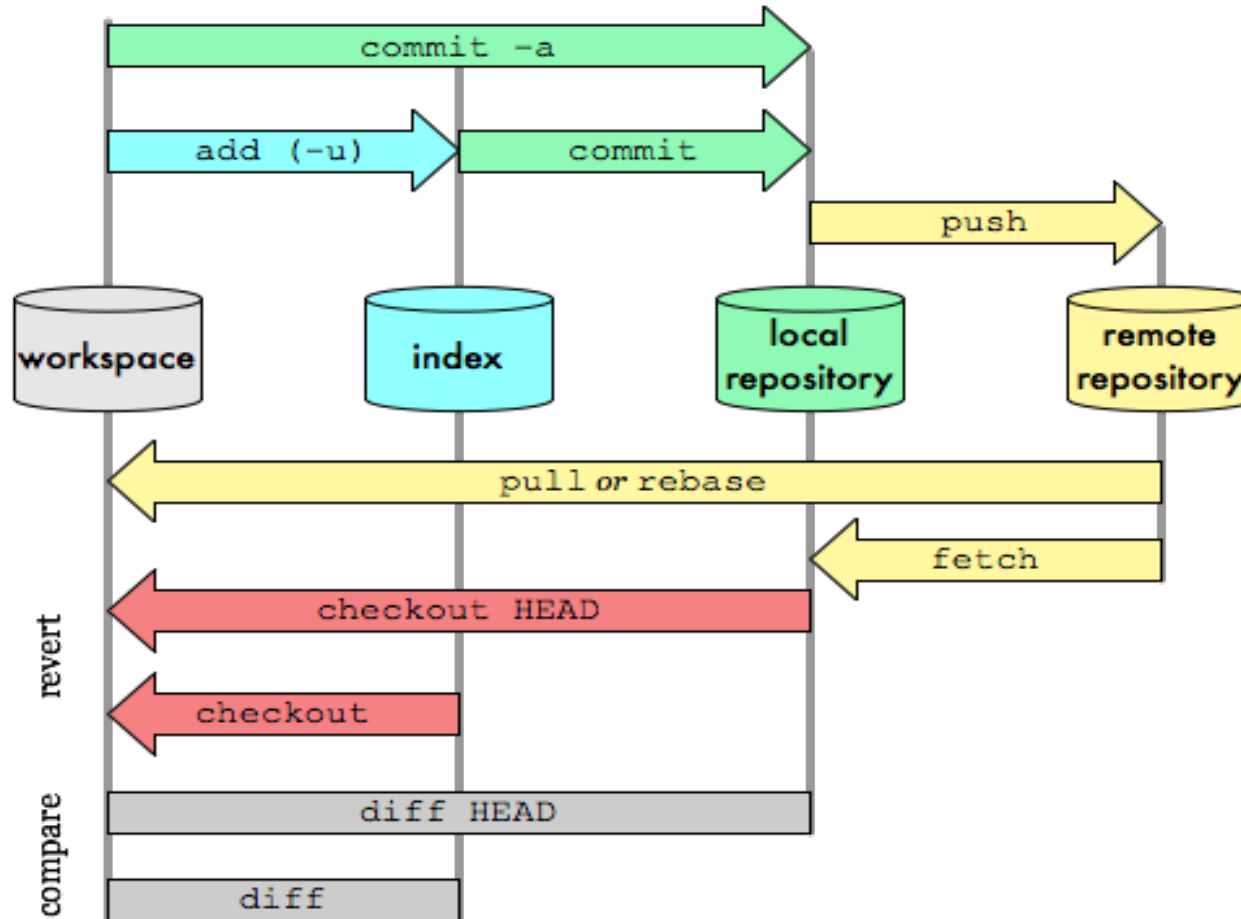
-> commit == fichiers versionnés

**Etc. ... mieux qu'un '*cp config.txt config.txt.init2*' non ?!**

# Git Workflow

## Git Data Transport Commands

<http://osteele.com>



# Dépôt local

## répertoire .git

à la racine

contient **tous** les commits

+ configuration locale

**à part pour partager tout se fait en local :**

manipulation des commits et des branches

consultation de l'historique (log, blame)

**dépôt équivalent au remote repository**

# Git – versioning local

## Gestion de version d'un répertoire avec des fichiers de configuration

Tracabilité

« Retours en arrière » possible

Etc.

**Etckeeper (debian) propose par exemple de versionner tout /etc en concordance avec apt.**

# Git – versioning test / prod

On valide sur un poste de test puis on « pousse » sur la production.

Avec Git, on peut maintenir des configurations spécifiques à chaque poste.

# Git pour faciliter les maj

Pour un outil géré via Git et publié sur github par exemple.

On le récupère par Git

On versionne par Git nos configurations internes

On fusionne (merge) nos modifications sur la version supérieure pour mise à jour.

# Inspiration, ressources ...

Wikipedia (pour l'historique)

Osteele.com (pour un schéma)

Une présentation ESUP de Pascal Rigaux (Paris1)

Github

<https://www.atlassian.com/git>

...





# EsupV4

**Paris, 12 Mai 2014**

# Recommandations (ou parti pris)

## Environnement

Debian - 32 Bits

Partie « Système OS »

- Par paquets (apt-get)
  - outils :
    - » emacs
    - » git-core
    - » gitg
  - services :
    - » Apache
    - » Ldap
    - » PostgreSQL

# Recommandations (ou parti pris)

## Partie Java

- JDK 1.6 32 - bits – install manuelle depuis distribution officielle Oracle (et non par paquet)
- /usr/local pour java, maven, ant
- /opt pour sources Esup4, tomcat, env .sh ...

## IDE / Editeur / GUI / ...

Ligne de commandes (pour git notamment)

Edition simple (VI, gedit, emacs ...)

Gitg pour visualisation GIT

Interfaces web EsupV4, Github, ...

# Recommandations (ou parti pris)

## On versionne en GIT

- Pour versioning interne des configurations
- Pour faciliter le travail sur N instances de dev, de preprod, de prod ...
- Et pour faciliter les mises à jour ...

## Configs, uPortal, ...

On tente d'être le plus efficace et pragmatique possible :

- on diverge le moins possible.
- on utilise les IHMs d'administration d'uPortal quand c'est possible (Portal Administration Tools), et quand ils fonctionnent.

# Machine virtuelle

Workshop monté sur une machine virtuelle

Object de la machine, install ESUPV4, cad héberger les services suivant :

- PostgreSQL
- Tomcat avec ESUP4 (et portlets)
- Apache en frontal du tomcat ESUP4

On y a installé également l'environnement SI « virtuelle » suivant :

- LDAP
- CAS
- Cela en plus des utilitaires git, emacs, java, etc.

password → esup

# Mise en place de la machine virtuelle

## Paquets

```
apt-get install emacs23-nox
```

```
apt-get install git-core gitg
```

```
apt-get install apache2
```

```
apt-get install postgresql
```

```
apt-get install slapd
```

## Manuellement

Jdk 1.6 ([oracle.com](http://oracle.com))

Apache-ant 1.8 ([apache.org](http://apache.org))

Apache-maven-3 ([apache.org](http://apache.org))

# LDAP

## Localhost 389

/home/esup/jxplorerer/jxplorerer.sh

## Comptes :

cn=admin,dc=esup-portail,dc=org / esup

uid=demo,ou=people,dc=esup-portail,dc=org / esup

uid=esup,ou=people,dc=esup-portail,dc=org / esup

# CAS

## Installation via cas-toolbox de Esup

CF /opt/cas :

- Cas-toolbox-svn config.properties :
  - ldap.univ.fr (modification de /etc/hosts)
  - cas.univ.fr (modification de /etc/hosts)
- Tomcat-cas (ln -s)
  - Install manuelle (DL de apache.org)
  - Confs par défaut → contexts dans webapps (ROOT pour CAS)
  - Conf port (server.xml) : 80xx → 90xx
  - /etc/init.d/tomcat-cas

Apache en proxy (ajp)

- Cas.univ.fr
- Ssl avec certificat wildcard (auto-signé ici)



# PostgreSQL

Host all all 127.0.0.1/8 password

Psql

- Create database esup4 ;
- Create user esup4 with password 'esup' ;
- Grant all on database esup4 to esup4 ;

# Apache

**ProxyPass AJP**

**Virtualhost SSL ent.univ.fr**

**Balancer pour load-balancing/failover**

- ent1.univ.fr, ent2.univ.fr pour ProxyCas
- sticky session avec cookie positionné par apache simplement

# Env – scripts - ...

## /opt/esup-env

env.sh pour JAVA\_HOME, JAVA\_OPTS (mémoire, var env, system property ...), M2\_HOME, PATH, etc.

scripts éventuels pour déploiement, init.d, ...

trustore java

...

## /opt/esup-env sous git

Source env.sh dans .bashrc utilisateur esup

# Récupération EsupPortail

→ **GIT**

```
git clone
```

```
https://github.com/EsupPortail/esup-uportal.git
```

```
cd esup-uportal/
```

```
git checkout -b esup4-univ uportal-4.0.11-esup-1
```

# Configuration

## Voir la doc / README

- build.properties
- filters/esup.properties
- pom.xml

# Déploiement

**Avec initialisation de la base :**

```
ant -Dmaven.test.skip=true clean initportal
```

# Lancement

`./startup.sh`

Ou mieux : `/etc/init.d/tomcat-esup restart`

# Gestion configurations

## On comit dans git :

```
git commit -a -m « EsupPortail de base avec  
cas+ldap »
```



# GIT - Application patch

## PostgreSQL : pb d'encodage

→ patch spécifique pour une installation sous postgresql

CF Doc Esup :

<http://www.esup-portail.org/pages/viewpage.action?pageId=269090823>

→ on réinitialise la base après : ant initdb