

Pràctica 4: Instal·lació de l'entorn de desenvolupament d'Integració Continua (IC) Jenkins

1- Creació d'un projecte Netbeans sincronitzat amb un dipòsit Github

1- Crea un dipòsit de nom **jenkins** de nivell d'accés **Públic** des del teu compte de **GitHub**.

2- Amb **Netbeans** crea un projecte de tipus **Java Application** i de nom **jenkins**. Canvia el codi font inicial pel codi font que trobaràs a:

```
wget http://www.collados.org/daw2/m08/uf4/Jenkins.java
```

NOTA: Canvia l'autor pel teu nom d'usuari de **GitHub**.

3- Crea un dipòsit local **Git** amb **Netbeans**. Fes un **Add** i un **Commit** (amb el missatge "Prova d'IC amb Jenkins") de tot el projecte **jenkins**. **Puja** el dipòsit **local** al dipòsit remot **GitHub**.

2 - Instal·lació de Jenkins en debian. Integració amb Git i Github

0.- Si tens instal·lat el contenidor web **tomcat7** (o posterior) escoltant pel port **8080**, atura'l previament.

1.- Descarrega l'arxiu de guió **inst_jenkins.sh** amb l'ordre:

```
wget http://www.collados.org/daw2/m08/uf4/inst_jenkins.sh
```

Afegeix-li permisos d'execució. Executa-ho (aquest procés pot durar uns minuts). Comprova que aquest script instal·la i posa en marxa l'entorn **IC Jenkins**.

2.- Comprova que una aplicació **java** que escolta pel port **8080** s'ha iniciat. Accedeix a l'aplicació des del navegador connectant-te a **http://localhost:8080**. Comprova que accedeixes a la **web d'administració de Jenkins**.

3- Validat dins de **Jenkins** amb la contrasenya d'administrador que trobaràs a **/var/lib/jenkins/secrets/initialAdminPassword**.

4- Instal·la els plugin **GitHub Plugin** (altres plugin seran instal·lat).

5- Crea un compte d'administrador amb el mateix nom i correu que l'usuari de **GitHub** que vas crear a la pràctica **m08uf4pr1a**. La contrasenya serà **fjeclot**.

6- A **Manage Jenkins (Configuració de Jenkins) --> Global Tool Configuration** afegeix la versions de **JDK** instal·lades al sistema. Per saber quines versions de **java** teniu disponibles, executeu: **update-alternatives --config java**. Mireu les captures d'imatge al final del document (ANNEX) per tenir una idea de com dur a terme la configuració. Potser has d'instal·lar el programari **openjdk**. Per instal·lar-lo, executa **aptitude install openjdk-7-jdk** o **openjdk-8-jdk** en funció de la versió de **java** seleccionada.

7- A **Manage Jenkins --> Global Tool Configuration**, Afegeix **Ant** com a eina per automatitzar el procés de construir el projecte. El nom serà **ant-1.9.4**, que és la versió actual per **debian 8.6.0** i a **ANT_HOME** el directori és **/usr/share/ant**. Mireu les captures d'imatge al final del document (ANNEX) per tenir una idea de com dur a terme la configuració. Potser has d'instal·lar el programari **ant** per fer aquest apartat. Per instal·lar-lo, executa **aptitude install ant**.

8- A **Manage Jenkins --> Configure System**, afegeix la configuració del servidor **SMTP**. Utilitza el servidor de gmail **smtp.gmail.com**. Comprova al final del document (ANNEX) com fer la configuració.

9- Reinicia **jenkins** i torna a validar-te.

3- Creaci d'un treball d'IC d'un projecte

1- Des de **Item nova (New item)**, crea una nou **treball** d'integració continua de nom nom **ic_jenkins** de tipus **free-style**.

2.- Escriu com **Descripció** del projecte o treball *"Prova d'IC amb Jenkins"*

3.- Selecciona **Git** com a sistema de control de version.

4.- Indica l'adreça URL del dipòsit del projecte **jenkins** que has creat dins de **Github** al primer apartat.

5- Afegeix les credencials del teu usuari de Github.

6- Indica que la branca a controlar és la **master**.

7- Configura el treball **ic_jenkins** perquè **cada 15 minuts, a totes les hores del dia, cada dia de la setmana, cada dia del mes i tots els mesos**. verifiqui si s'han fet canvis en el codi que disparin el procés de fer un "build" del projecte.

8- A la secció **Build**, indica **Ant --> Ant Versión --> ant-1.9.4**, com a programari per realitzar els builds (construccions del projecte).

9- A **Post-Build Actions** indica les adreces de correu electrònic a s'haurien d'enviar els missatges en cas d'errors en un **Build** del projecte.

10- Aplica els canvis. Desa el canvis.

4 – Construccions (build) d'un projecte. Comprovacions.

1- Des de la pàgina del projecte **ic_jenkins**, fes click a **Construir ara (Build Now)**. Comprova a **Build History** el resultat de fer una construcció del projecte. Comprova si ha estat un èxit o no (Comprova el color de la rodoneta i el símbol al costat de **Build History**). **NOTA:** El primer build pot trigar una estona més llarga que la resta de builds.

2- Accedeix al primer build i:

a) Comprova qui ha estat el responsable d'iniciar la construcció del projecte.

b) Entra a la consola de sortida (o console output). Comprova totes les accions realitzades. Comprova al final de tot que el resultat és BUILD SUCCESSFUL.

3.- Modifica el codi i afegeix darrera la línia 25 el següent codi:

```
bw.write(" Modificació del projecte jenkins");bw.newLine();
```

i després, fes un **Add** i un **Commit** de de la nova versió del projecte al dipòsit local Git. A continuació, fes un **Push** al dipòsit remot **Github**.

4- A contiució, fes una **construcció manual** del projecte **jenkins** des d'**ic_jenkins**. Accedeix al nou build (o construcció) i comprova que s'ha fet amb èxit i que queden reflexats els nous canvis. Comprova:

a) El responsable del Build (o construcció)

b) El canvis que hi ha respecte a l'últim build anant a "Changes"

c) Entra a la consola. Comprova totes les accions realitzades. Comprova al final de tot que el resultat és BUILD SUCCESSFUL.

5- Comprova que a les XX.00, XX.10h, XX.20h, XX.30..... es realitza una nova construcció del projecte. Comprova qui ha estat el responsable d'iniciar la construcció del projecte.

6- Modifica el codi i afegeix darrera la línia 25 el següent codi:

```
bw.write("    2a Modificació del projecte jenkins");bw.newLine()
```

i després, fes un **Add** i un **Commit** de de la nova versió del projecte al dipòsit local Git. A continuació, fes un Push al dipòsit remot **Github**.

7- Ara torna a fer una **construcció manual** del projecte **jenkins** des d'**ic_jenkins**. Comprova que queden reflectits els errors realitzats. Comprova el color de la rodona del **Build**, la icona de **Build History**.

8- Entra dins del Build amb errors i comprova a on s'han produït els errors anant a la consola. Mira els errors a la secció **-do-compile** al final de la consola de sortida. Comprova que també el resultat ha estat un **BUILD FAILED**. Comprova que t'indica també les línies amb errors.

9- Torna a **Netbeans**, comprova que els errors s'han trobat correctament. Soluciona els errors. Puja el nou projectes al Git local i a Github. Torna a fer un Build des d'**ic_jenkins** i comprova que no hi han errors.

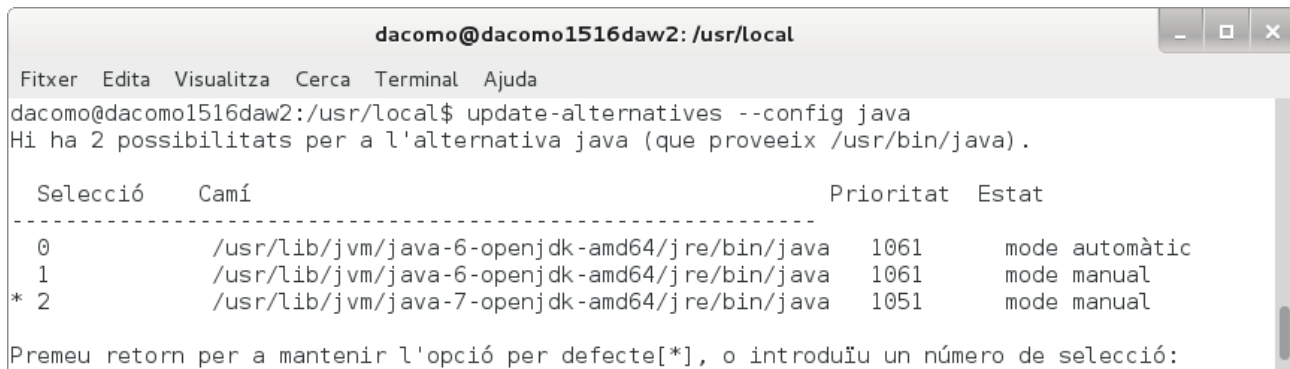
10- Comprova si t'han arribat missatges de correu pels **Build** amb errors.

Lliurament de la pràctica

- 1- Comprovació de la Instal·lació de **jenkins**.
- 2- Comprovació de la configuració de les tasca **jenkins**.
- 3- Execució d'un **build manual** del programa **jenkins** des del sistema d'**Integració Continua**.
- 4- Configuració d'un build automàtic.
- 5- Les comprovacions es faran els dies **15-2-17** i **20-2-17**.

ANNEX

1- Directori HOME de JAVA

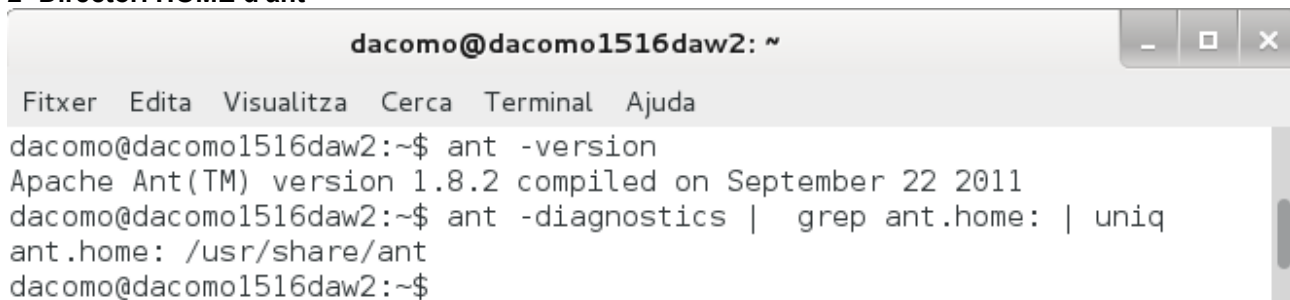


```
dacom@dacomo1516daw2: /usr/local
Fitxer Edita Visualitza Cerca Terminal Ajuda
dacom@dacomo1516daw2:/usr/local$ update-alternatives --config java
Hi ha 2 possibilitats per a l'alternativa java (que proveeix /usr/bin/java).

Selecció    Camí
-----
0           /usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java    1061    mode automàtic
1           /usr/lib/jvm/java-6-openjdk-amd64/jre/bin/java    1061    mode manual
* 2         /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java    1051    mode manual

Premeu retorn per a mantenir l'opció per defecte[*], o introduïu un número de selecció:
```

2- Directori HOME d'ant



```
dacom@dacomo1516daw2: ~
Fitxer Edita Visualitza Cerca Terminal Ajuda
dacom@dacomo1516daw2:~$ ant -version
Apache Ant(TM) version 1.8.2 compiled on September 22 2011
dacom@dacomo1516daw2:~$ ant -diagnostics | grep ant.home: | uniq
ant.home: /usr/share/ant
dacom@dacomo1516daw2:~$
```

3- Configuració JDK

JDK

Instal·lacions JDK

JDK

Nom

openjdk7

JAVA_HOME

/usr/lib/jvm/java-7-openjdk-amd64

Install automatically

Afegir JDK

Llista de JDK instal·lacions en aquest sistema

4- Configuració ANT

Ant

Instal·lacions Ant

Ant

Name

ant-1.8.2

ANT_HOME

/usr/share/ant

Install automatically

Afegir Ant

Llista de Ant instal·lacions en aquest sistema

5- Configuració del correu

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix

Use SMTP Authentication

User Name

dc1617daw2@gmail.com

Password

Use SSL

SMTP Port

465

Reply-To Address

Charset

UTF-8

Test configuration by sending test e-mail

Save

Apply

6- Algunes consells

- a) Comprova que **git** està instal·lat.
- b) És convenient instal·lar el paquet **openjdk-7-jdk** i/o **openjdk-8-jdk** .
- c) És convenient indicar al sistema que la **versió** del paquet **openjdk** que es vol utilitzar és la **7** o la **8**. Amb l'ordre **update-alternatives - -config java** es pot escollir la versió més adequada.
- d) Dins de **Netbeans** s'hauria d'indicar que el directori de la instal·lació de JDK és aquell que ha sortit quan s'ha executat l'ordre que es veu a la primera captura de pantall de l'annex.
- e) Instal·la **jenkins** després d'instal·lar i escollir la versió més adequada del paquet **openjdk**.