

## Pràctica 4: Integració Continua (IC) amb Jenkins

### 1- Creació d'un projecte dins de Github per poder ser controlat amb Jenkins

1- Clona el dipòsit <https://github.com/dc1617daw2/jenkins.git> dins de la teva màquina virtual. Dins del directori **jenkins** que s'ha creat, accedeix al fitxer **src/jenkins/Jenkins.java**. Modifica **Jenkins.java** fent que l'autor del projecte sigui el de teu compte de **GitHub**.

2- Des Fes un **Add** i un **Commit** (ho has de fer des de la carpeta arrel del projecte a on es troben build.xml, manifest.mf,...) del projecte escrivint el missatge "*Prova d'IC amb Jenkins*" i canviant l'autor del commit pel teu compte de correu utilitzat per treballar amb **Gitbhub** fent servir l'opció **--author** i amb la informació que trobaràs [aquí](#).

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

4- Esborra els **remote** antics que han quedat de fer el **clone**. Executa: **git remote rm origin**

5- Crea un nou **remote** amb l'adreça URL del teu projecte de **GitHub** tal i com s'indica a la pàgina del projecte de **GitHub** a la secció **..or push an existing repository from the command line**.

6- Puja el dipòsit local al dipòsit remot **GitHub** amb un **git push** tal i com s'indica a la pàgina del projecte de **GitHub** a la secció **..or push an existing repository from the command line**. Assegura't que al dipòsit remot s'han pujat els **2 commits** i **nbproject**, **src/jenkins**, **build.xml** i **manifest.mf**.

### 2- Instal·lació d'OpenJDK per desenvolupar i executar aplicacions Java sobre Debian Linux

Instal·la el Kit de desenvolupament Java (JDK) que inclou:

- a) L'entorn d'execució JRE (Java Runtime Enviroment) amb la màquina virtual java (JVM) i altres components necessaris per executar aplicacions Java dins del teu sistema operatiu
- b) Les eines de desenvolupament: biblioteques, documentació i utilitats (compiladors, interprets, gestors de paquets jar, etc....)

La versió open source de JDK s'anomena OpenJDK i per Debian Linux es pot utilitzar la versió 11 instal·lant el paquet **openjdk-11-jdk** i **altres paquets** . Per realitzar la instal·lació, executa:

```
aptitude install openjdk-11-jdk
```

### 3 - Instal·lació de Jenkins en Debian. Integració amb Git i Github

0- Abans de començar, executa amb privilegis de **root** l'ordre **netstat -atupn** i comprova que cap servei utilitza el port **8080/tcp**. Si hi ha cap servei que utilitza aquest port, atura'l.

1.- Crea una carpeta de nom **temp** dins del teu directori personal. Accedeix a **temp** i amb l'ordre **wget** descarrega l'arxiu de guió **inst\_jenkins.sh**:

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

Afegeix-li permisos d'execució. Executa-ho amb privilegis d'usuari **root**. En funció de la velocitat de la connexió a internet aquest procés pot durar des d'uns 30 segons a un número indeterminat de minuts.

2.- Comprova que aquest script instal·la el paquet **jenkins** i posa en marxa l'entorn **IC Jenkins**. Executa:  
**aptitude search jenkins** → Comprovació que el paquet de programari ha estat instal·lat  
**systemctl status jenkins** → Comprovació que el sistema d'IC jenkins està funcionant.

3.- El sistema d'IC **Jenkins** és una aplicació **java** que obre el port de xarxa **8080/tcp**. Comprova que una aplicació **java** que escolta pel **port 8080** s'ha iniciat i està en execució dins del teu sistema. Per comprovar-ho, executa amb privilegis de **root** l'ordre **netstat -atupn**.

4.- Accedeix a l'aplicació des del navegador connectant-te a **http://localhost:8080**. Comprova que accedeixes a la **web d'administració de Jenkins**.

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

6.- Instal·la els plugins recomanats. Això pot trigar uns minuts.

7.- Crea un compte d'administrador que tingui com a **Username** el teu nom **d'usuari de Github** i **E-mail address** el correu de l'usuari de **Github**. La contrasenya serà **FjeClot21#**. Com a **Full name** escriu el nom de l'usuari de **Github**.

#### **4- Configuració bàsica de Jenkins per fer l'integració continua d'un projecte de Java que utilitza Git com a SCV i Github com a dipòsit remot.**

1.- Accedeix a **Manage Jenkins (Configuració de Jenkins) → Manage Plugins → Installed**, i assegura't que a llista de plugins instal·lats trobes els plugins **Ant plugin**, **Git plugin** i **GitHub plugin**. En cas de no ser-hi, instal·la'ls anant a la secció de **Available** de la pàgina de gestió de plugins.

2.- Dins de **Manage Jenkins** accedeix a **Global Tool Configuration** i indica quina és la versió de **JDK** instal·lada al sistema. Per saber quines versions de **java** teniu disponibles i quin directori es troba instal·lada. Executeu: **update-alternatives --config java** (Nota: el directori és tot el que hi ha abans de **/bin/java**). La secció **JDK** serà alguna cosa similar a això:

JDK

JDK installations

Add JDK

JDK

Name

JAVA\_HOME

Install automatically

Delete JDK

Add JDK

List of JDK installations on this system

3.- Dins de **Manage Jenkins** accedeix a **Global Tool Configuration** i indica a on est troba el programa **Git**. Per saber la versió de **Git** executa **git --version** i per saber el camí cap a l'executable de **Git** executa **whereis git**. La secció serà quelcom similar a això:

Git

Git installations

Git

Name

Path to Git executable

Set path to git executable. This can be just "git" or complete path. (from Jenkins Git client plugin)

Install automatically

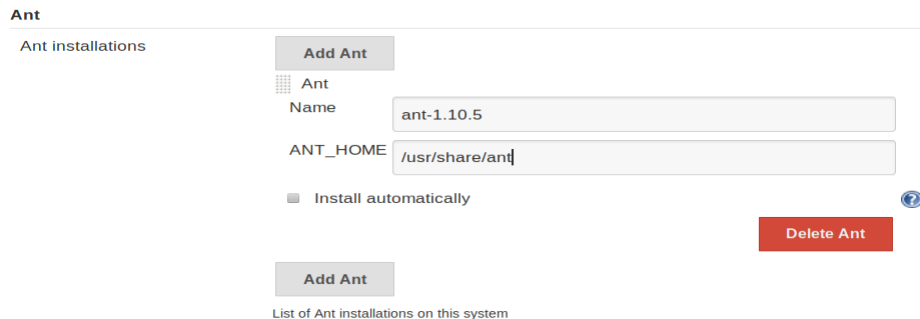
Delete Git

4.- A **Configuració de Jenkins (Manage Jenkins)** --> **Global Tool Configuration**, Afegeix **Ant** com a eina per automatitzar el procés de construir el projecte. Per saber la versió d'**Ant** instal·lat al teu sistema executa i el directori **ANT\_HOME** a on s'ha instal·lat executa:

**ant -version** --> Mostra la versió d'**Ant**. En principi, la vostra versió hauria de ser la **1.10.5**.

**ant -diagnostics | grep ant.home:** | **uniq** --> Mostra el directori **ANT\_HOME**.

Un cop realitzades les comprovacions, la configuració d'**Ant** hauria de ser quelcom similar a això:



**NOTA:** Si **ant** no ha estat instal·lat, executa **aptitude install ant**. Comprova posteriorment que s'ha instal·lat **ant** i **ant-optional**. El programa **ant** fa una feina similar a **Make** per programes en **C** que varem veure l'any passat però orientat a aplicacions **Java**.

5.- A **Configuració de Jenkins (Manage Jenkins)** --> **Configure System** → **Jenkins Locations** → **System Admin e-mail address**, afegeix la configuració de la capçalera de les notificacions que via E-Mail envia Jenkins a responsables de projecte. Escriu: "Jenkins Daemon <la teva adreça de correu pel teu usuari de Github>". Per exemple: "Jenkins Daemon <[dacomo2021daw2@protonmail.com](mailto:dacomo2021daw2@protonmail.com)>" en el meu cas.

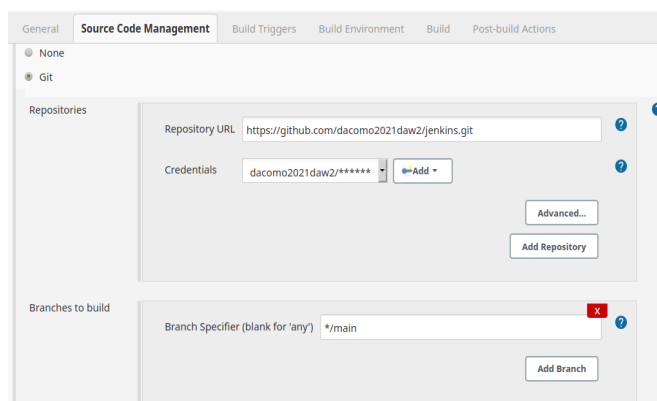
6.- Surt de la web d'administració de **Jenkins**. Reinicia **jenkins** executant **systemctl restart jenkins**, i torna a validar-te dins de la web d'administració. Ara s'ha finalitzat el procés de configuració inicial.

### 3- Creació d'un treball d'IC d'un projecte

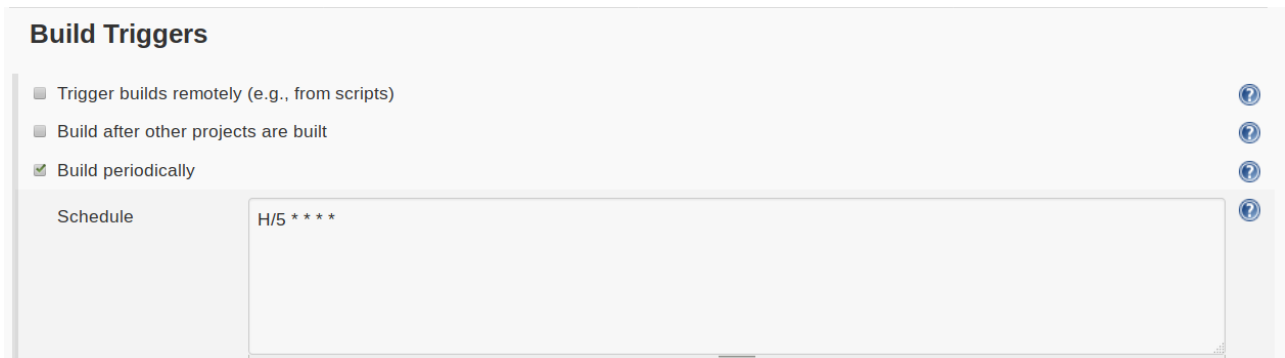
1- Des de **New item(item nova)**, crea una nou treball d'integració continua de nom **ic\_jenkins** de tipus **Freestyle**.

2.- Escriu com **Descripció** del projecte o treball *"Prova d'integració continua amb Jenkins sobre el projecte de nom jenkins que es troba al meu compte de Github"*

3.- Selecciona a la secció **Source Code Management (Gestor de Codi Font)** l'opció **Git** com a sistema de control de versions. Indica l'adreça URL del dipòsit del projecte **jenkins** que has creat dins de **Github** al primer apartat, i afegeix les credencials del teu usuari de **Github** (només cal nom i contrasenya) per permetre que l'eina **Jenkins** pugui accedir-hi i fer **Builds** del projecte. Indica que la branca a controlar és la branca **main**. Un cop configurat ha de quedar així (però a **Credentials** ha de sortir el teu nom d'usuari de Github):



4.- Configura el treball **ic\_jenkins** per realitzar un **Build** del projecte que es troba al **Github** amb una periodicitat de **5 minuts, a totes les hores del dia, cada dia de la setmana, cada dia del mes i tots els mesos de l'any**. Per realitzar aquesta configuració has de seleccionar **Build Triggers** --> **Build Periodically** --> **Schedule**. Per la periodicitat indicada, ha de posar:



5.- A la secció **Build**, indica **Add build step** --> **Invoke Ant** --> **Ant Versión** --> **ant-1.10.5**, com a programari per realitzar els Builds.

6.- A **Post-Build Actions**, indica dins de **Add post-build action** --> **E-mail Notification** --> **Recipients** una adreça de correu a la qual s'hauria d'enviar un correu si es produeix un **Build** amb errors o qualsevol altra problema.

7.- Aplica i Desa tots els canvis. Ara ja s'ha creat un treball d'integració continua de nom **ic\_jenkins** per realitzar el seguiment del projecte **jenkins** que es troba a **Github**.

#### **4 – Construccions (build) d'un projecte. Comprovacions manuals i automàtiques.**

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. Si ha estat un **èxit**, el color de la rodoneta del build realitzat ha de ser color **blau**, en cas contrari sera **vermell**. En el cas de que sigui blau, vol dir que Jenkins ha pogut accedir sense problemes al projecte que es troba a **Github** i que l'ha pogut compilar i no ha trobat cap error.

2.- Accedeix al **primer Build** i comprova:

a) Que el **Build** ha estat realitzat manualment per l'usuari mirant el rombe de color taronja.

b) Entra a la **consola de sortida** (o console output). Comprova que:

**b.1)** S'ha accedit al dipòsit de Github i s'ha clonat amb un **clone** i s'ha baixat el codi del dipòsit amb un **fetch**.

**b.2)** Comprova que s'ha compilat el codi a les seccions **-do-compile**, **-do-jar-jar** i **-javadoc-build** i que el resultat final és un **BUILD SUCCESSFUL**.

3.- Dins del directori **jenkins** que s'ha creat, accedeix al fitxer **src/jenkins/Jenkins.java**. Modifica **Jenkins.java** i **esborra el punt i coma que hi ha al final de la línia 5** de manera que el codi sigui aquest:

```
package jenkins
```

Després, fes un **Add** i un **Commit** (ho has de fer des de la carpeta arrel del projecte a on es troben build.xml, manifest.mf,...). Al comentari del Commit escriu **"Primera modificació de la prova d'IC amb Jenkins"**. A continuació, fes un **Push** al dipòsit remot **Github**. Comprova que s'ha pujat la darrera versió del projecte des de la web de **Github**.

- 4.- Ara torna a fer una **construcció manual** del projecte **jenkins** des d'**ic\_jenkins**. Comprova que:
- Quedan reflectits els errors realitzats perquè el color de la rodona del darrer **Build** és de color **vermell**.
  - Comprova que s'ha compilat el codi i que el resultat final és un **BUILD FAILED** a la secció **-do-compile**.
  - Comprova que la secció **-do-compile** mostra la línia a on es troba l'error i el tipus d'error.

5- Soluciona l'error creat a l'apartat 3 i torna a fer un **Add** i un **Commit** amb el comentari “*Segona modificació 2 de la prova d'IC amb Jenkins*”. Fes un nou **Push** al dipòsit remot **Github** i comprova novament que s'ha pujat la darrera versió del projecte des de la web de **Github**. Torna a fer un **Build** i comprova que surt de color **blau** novament.

6.- Deixa al sistema **Jenkins** funcionant i comprova que es realitzen els Builds planificats. Deixa que es facin 5 o 6 **Builds** per veure que tot funciona correctament. Entra dins de qualsevol d'aquest Builds automàtics i comprova que al mig de la pàgina surt una icona **taronja** en forme de **rombe** que diu que el Build va ser “**iniciat pel temporitzador**” o **Started by timer**.

### **Lliurament de la pràctica**

- 1- Comprovació de la configuració de les tasca **ic\_jenkins** dins de **Configura**.
- 2- Execució d'un **build manual sense errors**.
- 3- Comprovació de que s'ha realitzat el **builds amb errors** indicat als **punt 4** de la **secció 4**.
- 4- Veure una llista de múltiples **Builds** realitzats cada **2 minuts** de **manera automàtica iniciat pel timer**.
- 5- Les comprovacions es començaran a fer el dia **1-2-21**.