

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- Fes una Neteja i Construcció del projecte (Clean and Build) per comprovar que el codi és correcte i no té errors.

3- **Inicialitza** un dipòsit local **Git** amb **Netbeans**. Selecciona tot el projecte **jenkins** sencer. Fes un **Add** i un **Commit** escrivint el missatge "*Prova d'IC amb Jenkins*" i canviant **Author** i **Committer** pel teu compte de correu utilitzat per treballar amb **GitHub**. **Puja** el dipòsit **local** al dipòsit remot **GitHub**. Assegura't que al dipòsit remot s'han pujat **nbproject**, **src/jenkins**, **build.xml** i **manifest.mf**.

4-

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

1.- Si tens instal·lat i en execució el contenidor web **tomcat7** (o posterior) de **NetBeans** escoltant pel port **8080**, atura'l, i tanca el projecte **jenkins** i **NetBeans**. Assegura't en tot cas que cap servei escolta pel port **8080**.

2.- Com usuari **root** 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**.

3.- Comprova que una aplicació **java** que escolta pel port **8080** s'ha iniciat. Executa l'ordre:

```
ss -atupn | grep -E '(Netid|8080)'
```

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 el plugins recomanats o si no vols instal·lar-los tot, com a mínim instal·la **GitHub Plugin** (altres plugin seran instal·lats).

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

8.- A Configuració de Jenkins (Manage 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**. La secció **JDK** ha de quedar així:

JDK

Instal·lacions JDK

JDK

Nom

JAVA_HOME

Install automatically

Lista de JDK instal·lacions en aquest sistema

NOTA: Si el **JDK** no està instal·lat, executa **aptitude install openjdk-8-jdk**. Comprova posteriorment que s'ha instal·lat **openjdk-8-jdk**, **openjdk-8-jdk-headless**, **openjdk-8-jre** i **openjdk-8-jre-headless**.

9.- 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 **1.9.9**.
ant -diagnostics | grep ant.home: | **uniq** --> Mostra el directori **ANT_HOME**

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

Ant

Instal·lacions Ant

Ant

Name

ANT_HOME

Install automatically

Lista de Ant instal·lacions en aquest sistema

NOTA: Si **ant** no ha estat instal·lat, executa **aptitude install ant**. Comprova posteriorment que s'ha instal·lat **ant** i **ant-optional**.

10.- A Configuració de Jenkins (Manage Jenkins) --> Configure System, afegeix la configuració del servidor **SMTP**. Utilitza el servidor de gmail **smtp.gmail.com**.

The screenshot shows the 'E-mail Notification' configuration page in Jenkins. The 'SMTP server' is set to 'smtp.gmail.com'. The 'Default user e-mail suffix' is empty. The 'Use SMTP Authentication' checkbox is checked. The 'User Name' is 'dc1718daw2@gmial.com'. The 'Password' field is masked with dots. The 'Use SSL' checkbox is checked. The 'SMTP Port' is '465'. The 'Reply-To Address' is empty. The 'Charset' is 'UTF-8'. There is a checkbox for 'Test configuration by sending test e-mail' which is currently unchecked.

Evidentment, has de canviar el valor d'**User Name** pel teu compte de correu de **gmail**, que és el que utilitzes per **Github**.

11.- Surt de la web d'administració de **Jenkins**. Reinicia **jenkins** executant **service jenkins restart**, 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

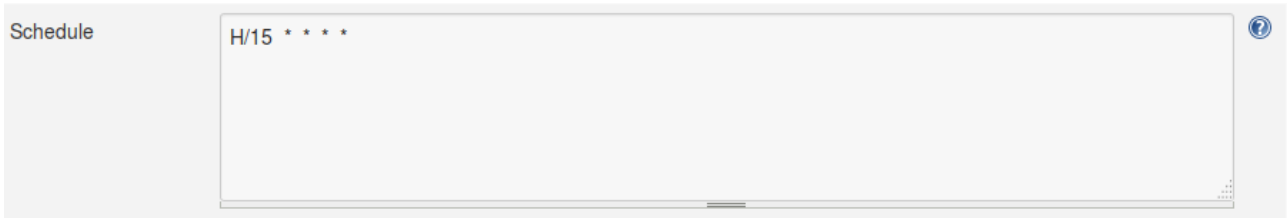
1- Des de **Item nova (New item)**, crea una nou **treball** d'integració continua de nom nom **ic_jenkins** de tipus **free-style**.

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

3.- Selecciona a la secció **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** per permetre que l'eina **Jenkins** pugui accedir-hi i fer **Builds** del projecte. Indica que la branca a controlar és la branca **master**. Un cop configurat ha de quedar així (però a **Credentials** ha de sortir el teu nom d'usuari de Github):

The screenshot shows the 'Gestor de Codi Font' configuration page in Jenkins. The 'None' radio button is selected, and the 'Git' radio button is also selected. Under 'Repositories', the 'Repository URL' is 'https://github.com/dc1718daw2/jenkins.git'. The 'Credentials' dropdown is set to 'dc1718daw2/*****'. There are 'Avançat...' and 'Add Repository' buttons. Under 'Branches to build', the 'Branch Specifier (blank for 'any')' is '*/master'. There is a red 'X' icon and an 'Add Branch' button.

4.- Configura el treball **ic_jenkins** per realitzar un **Build** del projecte que es troba al **Github** amb una periodicitat de **15 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-1.9.9**, 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**.

2.- Accedeix al primer Build. 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.- Amb **NetBeans**, modifica el codi del projecte **jenkins**, i afegeix darrera la línia 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. Al comentari del Commit escriu *"Modificació 1 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.

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 canvis que hi ha respecte a l'últim build anant a "Canvis" (Changes)

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

5.- 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. Al comentari del Commit escriu *"Modificació 2 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.

6.- Ara torna a fer una **construcció manual** del projecte **jenkins** des d'**ic_jenkins**. Comprova que queden reflectits els errors realitzats perquè el color de la rodona del darrer **Build** és de color **vermell**.

7.- 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.

8.- Torna a **Netbeans**, comprova que els errors indicats per Jenkins són realment els que té el projecte. Soluciona els errors. i després, fes un **Add** i un **Commit** de de la nova versió del projecte al dipòsit local Git. Al comentari del Commit escriu *"Modificació 3 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.

9.- Torna a fer un **Build** des d'**ic_jenkins** i comprova que no hi han errors.

10.- Deixa al sistema **Jenkins** funcionant i després d'un parell d'hores, comprova que cada 15 minuts es realitza una nova construcció del projecte.

Lliurament de la pràctica

1- Comprovació de la Instal·lació de **jenkins**.

2- Comprovació de la configuració de les tasca **ic_jenkins**.

3- Execució d'un **build manual** del programa **jenkins** des del sistema d'**Integració Continua**.

4- Comprovació de la **configuració** per realitzar **build automàtics**. Comprovació que s'han realitzar un **mínim de 10 builds automàtics**.

5- Comprovació de **builds amb errors**.

6- Les comprovacions es faran els dies **26-1-18** i **2-2-18**.