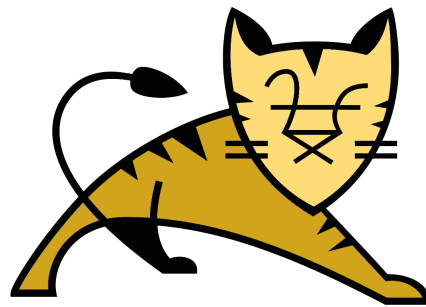


Instalación de Jenkins y Tomcat



Jenkins



Apache Tomcat

Román Millán Díaz

Índice

Índice	2
Instalación y Configuración	3
Instalación de Deploy to Container (Plugin)	3
Instalación de Maven (Plugin)	5
Configurar Proyecto en Jenkins	6
Configurar TomCat	11

Instalación y Configuración

Instalación de Deploy to Container (Plugin)

Para hacer la integración de nuestro proyecto con Jenkins lo primero que tenemos que hacer es instalar una serie de plugins en nuestro Jenkins.

El primero que vamos a instalar es **Deploy to Container**

Installation options

1. Using the GUI: From your Jenkins dashboard navigate to **Manage Jenkins > Manage Plugins** and select the **Available** tab. Locate this plugin by searching for *deploy*.
2. Using the CLI tool:

```
jenkins-plugin-cli --plugins deploy:1.16
```
3. Using direct upload. Download one of the [releases](#) and upload it to your Jenkins instance.

Nos vamos a nuestro host de Jenkins y a la pestaña Manage Jenkins.

Jenkins

Search (CTRL+K)

2

1

admin

log out

Dashboard > Manage Jenkins

+ New Item

People

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Manage Jenkins

New version of Jenkins (2.390) is available for [download](#) ([changelog](#)).

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agent

Set up cloud

Dismiss

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as BUILD_URL.

Dismiss

Please provide an accurate value in [Jenkins configuration](#).

System Configuration

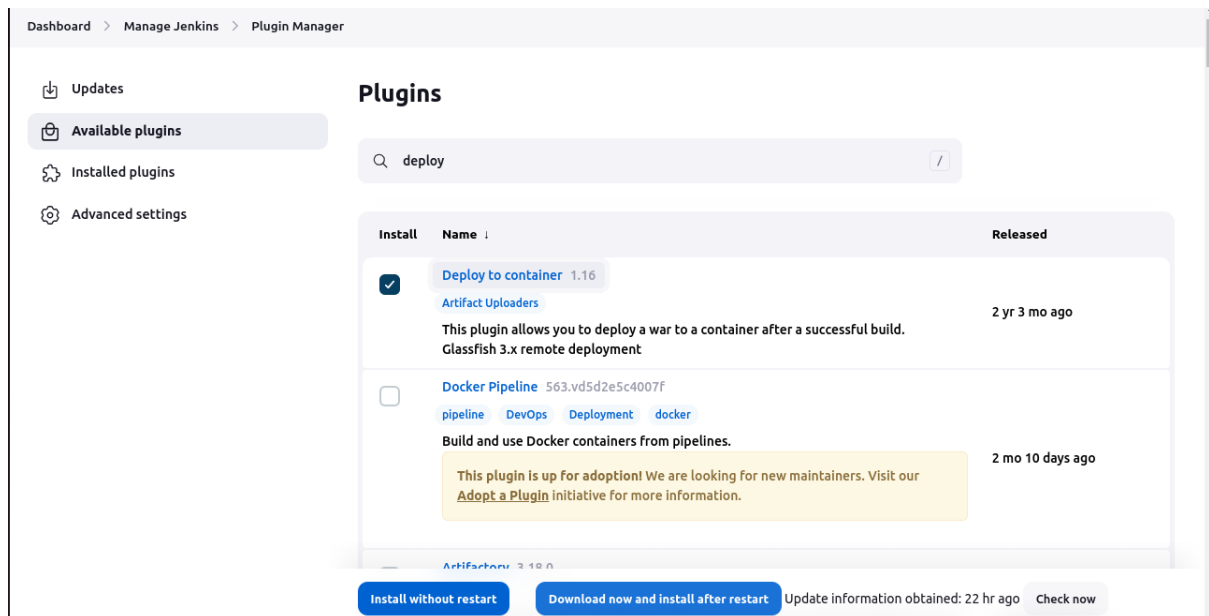
Configure System

Configure global settings and paths.

Global Tool Configuration

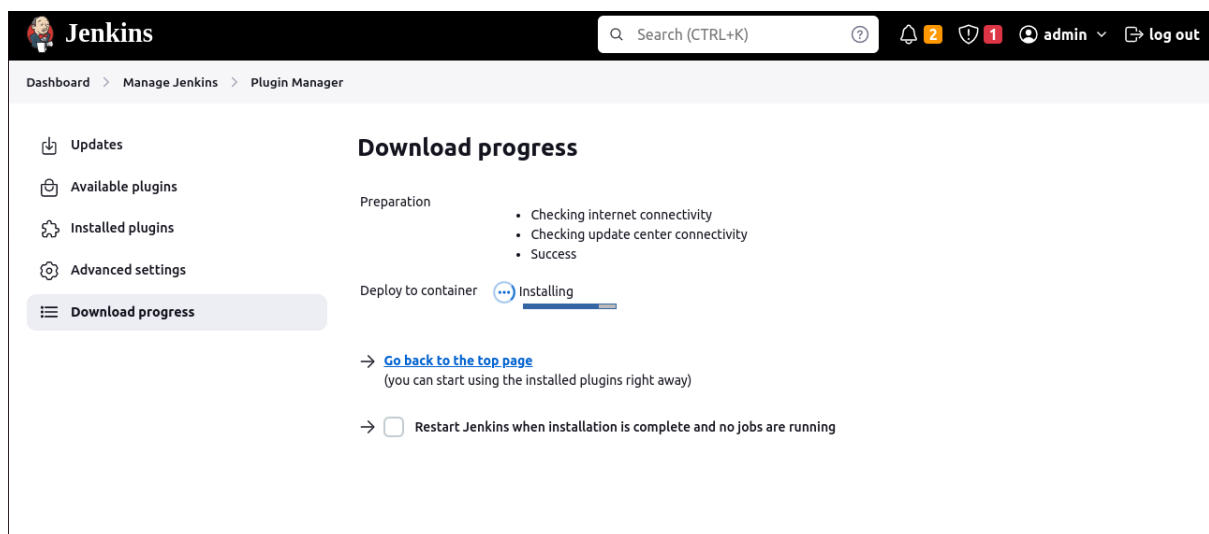
Configure tools, their locations and automatic installation.

Después vamos a Available plugins y buscamos el plugin **Deploy to container**.



Lo descargamos dando al botón **install without restart** (instalar sin reiniciar). También se puede con el botón download now and install after restart (más automático). En mi caso lo haré manual.

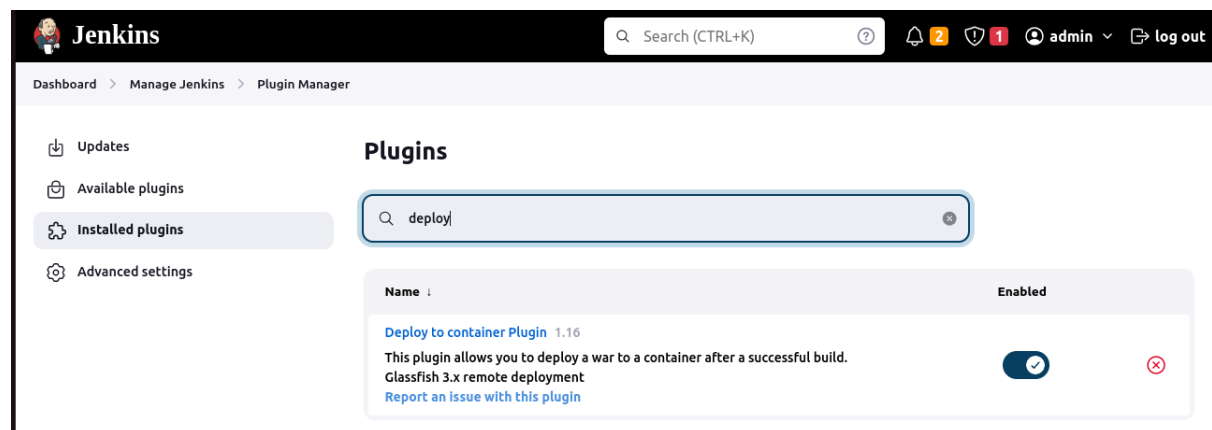
Una vez que se nos haya instalado correctamente, tenemos que reiniciar manualmente el Jenkins



Como podemos ver en la siguiente imagen tenemos que parar nuestro docker-compse e iniciarlo de nuevo.

```
Stopping jenkins_managed_by_compose ... done
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/2do Trimestre/Jenkins$ doc
ker-compose up
Starting jenkins_managed_by_compose ... done
Attaching to jenkins_managed_by_compose
jenkins_managed_by_compose | Running from: /usr/share/jenkins/jenkins.war
jenkins_managed_by_compose | webroot: /var/jenkins_home/war
jenkins_managed_by_compose | 2023-02-10 07:33:00.534+0000 [id=1] INFO w
instone.Logger#logInternal: Beginning extraction from war file
jenkins_managed_by_compose | 2023-02-10 07:33:00.616+0000 [id=1] WARNINGGo
.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
jenkins_managed_by_compose | 2023-02-10 07:33:00.697+0000 [id=1] INFO o
rg.eclipse.jetty.server.Server#doStart: jetty-10.0.13; built: 2022-12-07T20:13:2
0.134Z; git: 1c2636ea05c0ca8de1ffd6ca7f3a98ac084c766d; jvm 11.0.18+10
jenkins_managed_by_compose | 2023-02-10 07:33:01.055+0000 [id=1] INFO o
.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not f
ind org.eclipse.jetty.jsp.JettyJspServlet
jenkins_managed_by_compose | 2023-02-10 07:33:01.130+0000 [id=1] INFO o
```

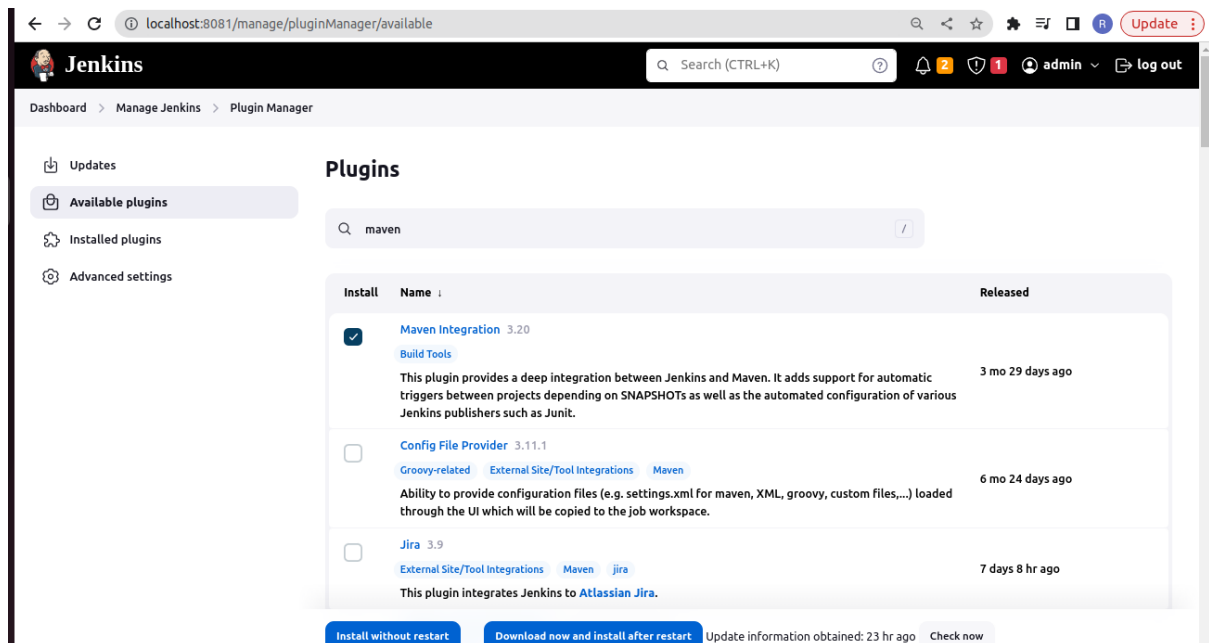
Una vez reiniciado podemos ir a **Manage Jenkins > Plugin Manager > Installed plugins**. Y buscar Deploy to container Plugin, para ver que está instalado y activado.



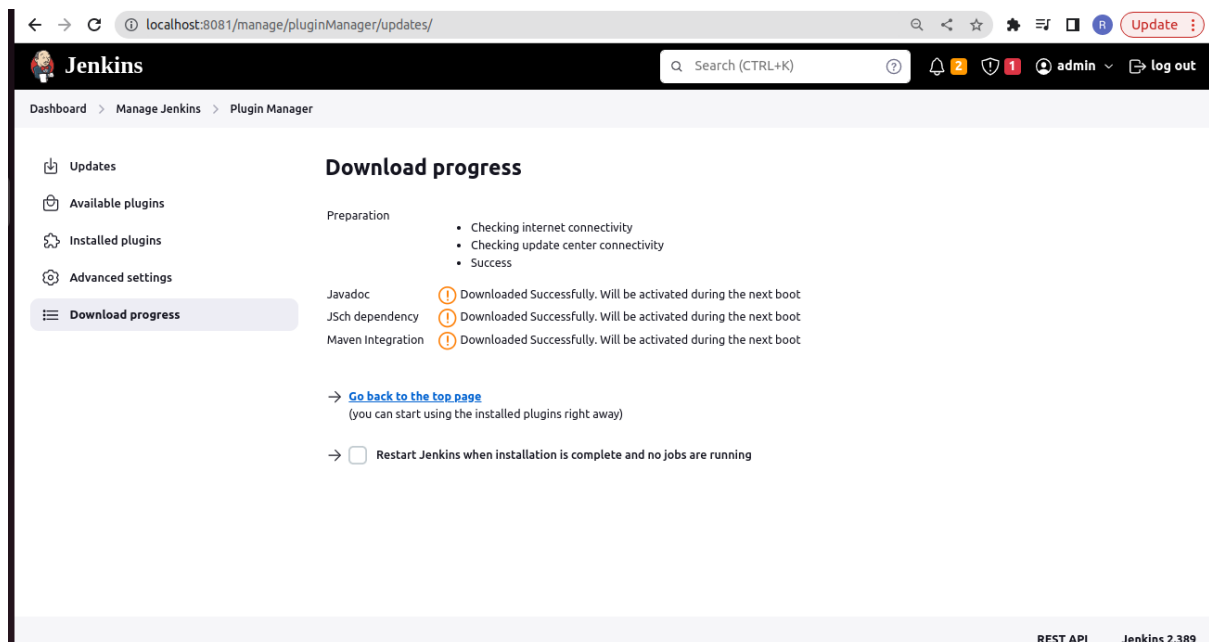
Instalación de Maven (Plugin)

Ahora una vez que tengamos el plugin Deploy to container Plugin instalado y activado, tenemos que instalar otro plugin en nuestro Jenkins llamado: **Maven integration**.

Al igual que el anterior plugin que acabamos de instalar, hacemos lo mismo con este mismo plugin.

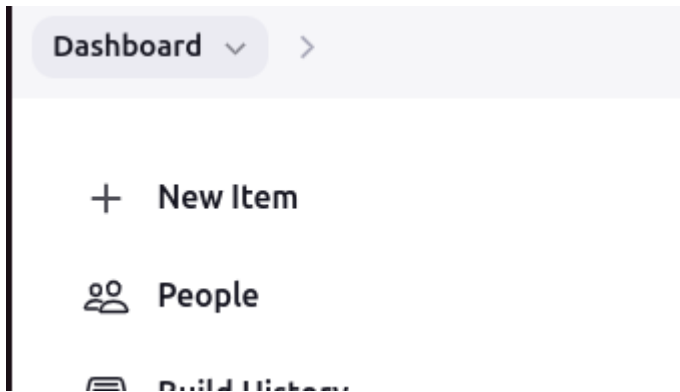


Una vez descargado e instalado , con su obligatorio reinicio ya tendremos todos los plugins que nos hace falta para seguir con la integración de nuestro proyecto en Jenkins

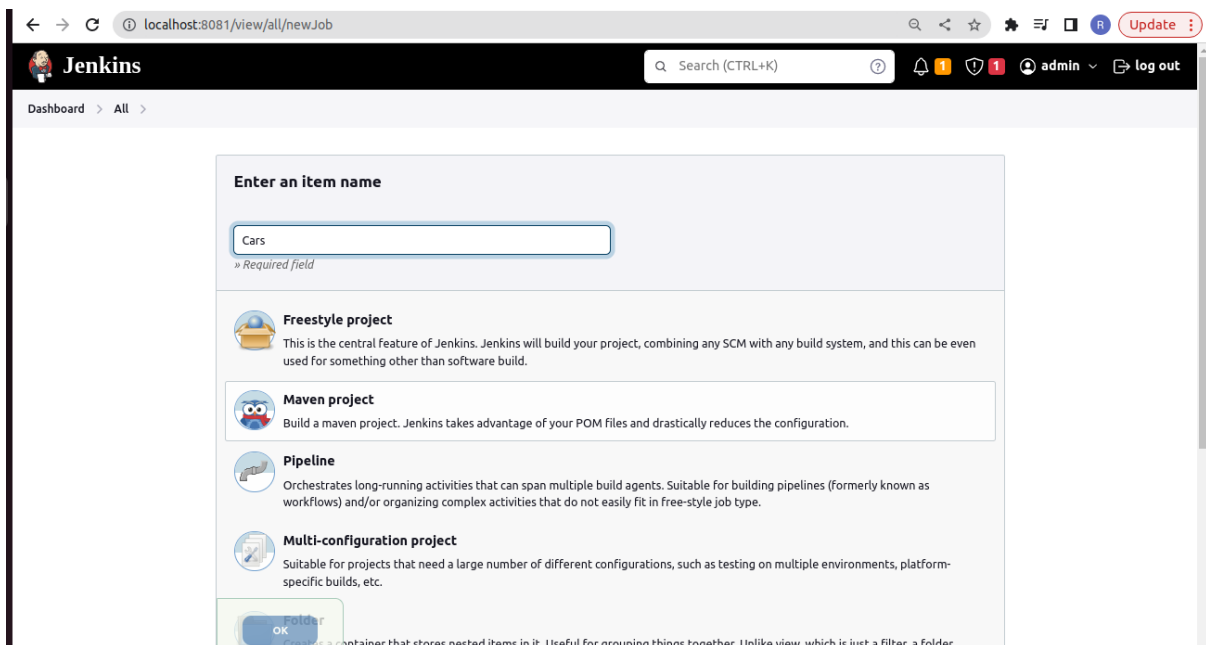


Configurar Proyecto en Jenkins

Ahora nos vamos a New Item, para crear un nuevo proyecto el cual después configuraremos para integrar nuestro proyecto de manera correcta.

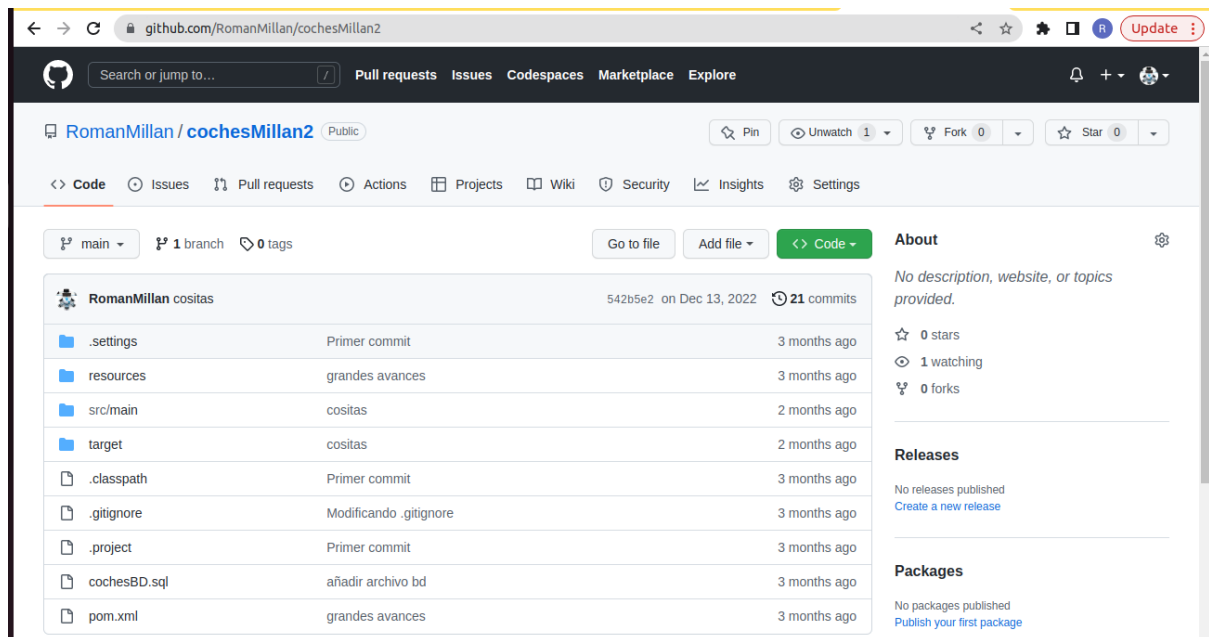


Lo primero que tenemos que hacer es poner un nombre a nuestro nuevo objeto que contendrá nuestro proyecto integrado en Jenkins.
Después le damos a **Maven Project**

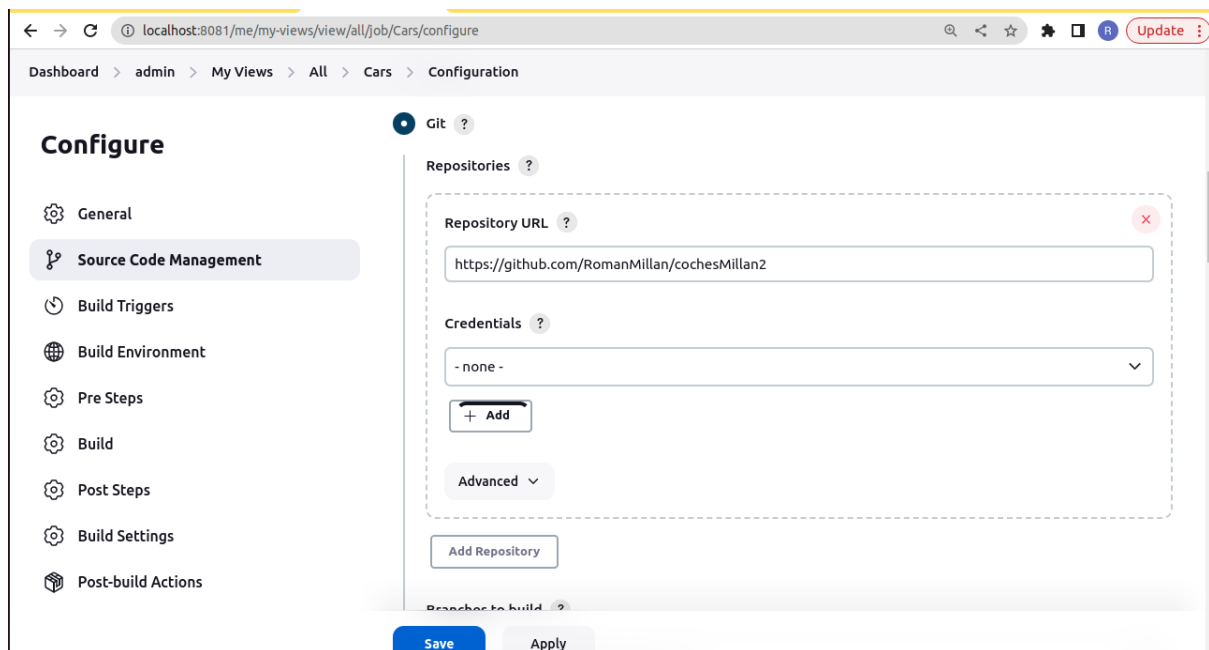


Y se nos abrirá una pestaña de configuración para nuestro proyecto.

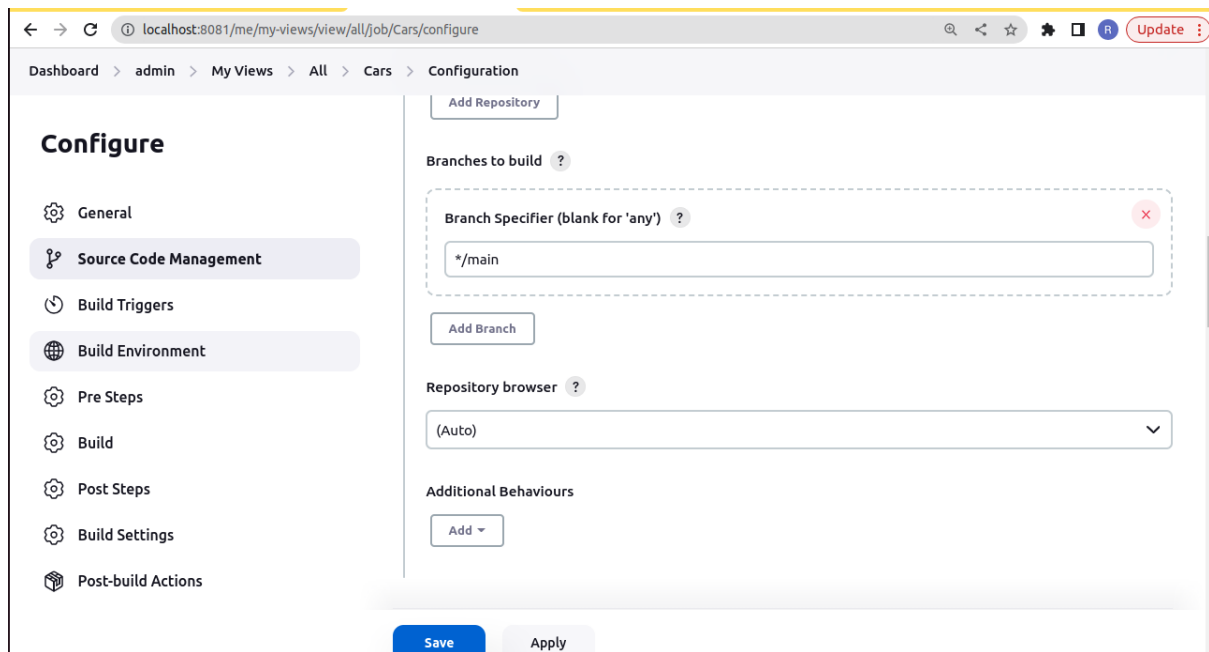
Una vez en la pestaña de configuración, lo primero que tenemos que hacer es ir a la pestaña **Source Code Management** y marcar la opción de **Git**. En repositories > Repository URL tendremos que escribir la url de nuestro repositorio a github donde tengamos subido el proyecto.



Como podemos ver en las imágenes, Repository URL tiene que coincidir perfectamente con la url de nuestro github.



Lo siguiente es ir a: **Branches to build > Branch Specifier (blank for 'any')**. Aquí tenemos especificar cuál es nuestra rama principal, donde buscará Jenkins para ejecutar el proyecto. En mi caso es ***/main**

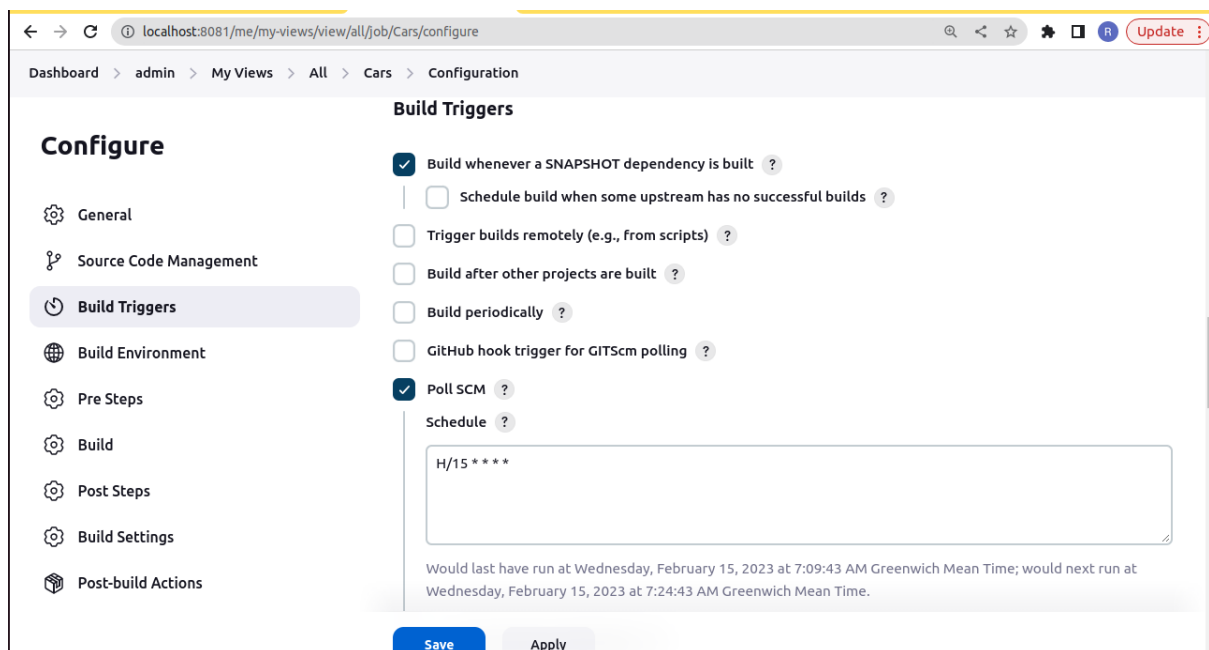


Después no vamos a Build Triggers y tenemos que activar dos pestañas.

- Build whenever a SNAPSHOT dependency is built
- Poll SCM

En Poll SCM tenemos que especificar con cuánta frecuencia Jenkins va a inspeccionar nuestro repositorio para actualizarlo.

En mi caso he puesto cada 15 minutos.



En **Pre Steps > Build > Root POM**. tenemos que escribir la ruta de nuestro pom.xml

The screenshot shows the Jenkins web interface at `localhost:8081/job/Cars/configure`. The breadcrumb navigation is `Dashboard > Cars > Configuration`. On the left, a sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps (highlighted), Build, Post Steps, Build Settings, and Post-build Actions. The main content area is titled 'Configure' and has an 'Add pre-build step' button. It is divided into three sections: 'Build' with a 'Root POM' field containing 'pom.xml' and a 'Goals and options' field; 'Post Steps' with an 'Advanced' dropdown; and 'Save' and 'Apply' buttons at the bottom.

Ya le podemos dar al botón de save (guardar).

En **Dashboard > Global Tool Configuration**: tenemos que configurar Maven; Añadimos **Add Maven > Maven**.

Tendremos que nombrar el maven y marcar **install automatically** y elegir la versión de nuestro maven.

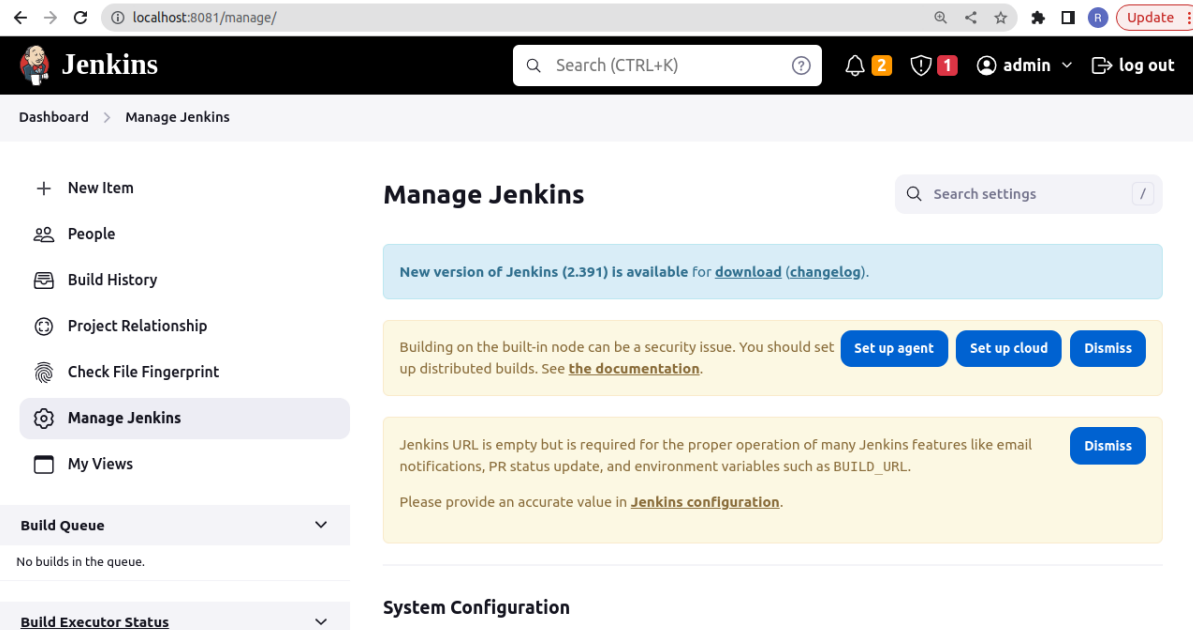
The screenshot shows the Jenkins 'Global Tool Configuration' page. The breadcrumb navigation is `Dashboard > Global Tool Configuration`. At the top, there is an 'Add Maven' button. Below it, a dashed box contains the configuration for a tool named 'Maven'. The 'Name' field is filled with 'MavenName' and has a red 'Required' error message below it. The 'Install automatically' checkbox is checked. Underneath, there is a section for 'Install from Apache' with a 'Version' dropdown menu set to '3.9.0'. At the bottom of the dashed box is an 'Add Installer' button. Below the dashed box, there are 'Save' and 'Apply' buttons.

Una vez guardado ya lo tenemos configurado.

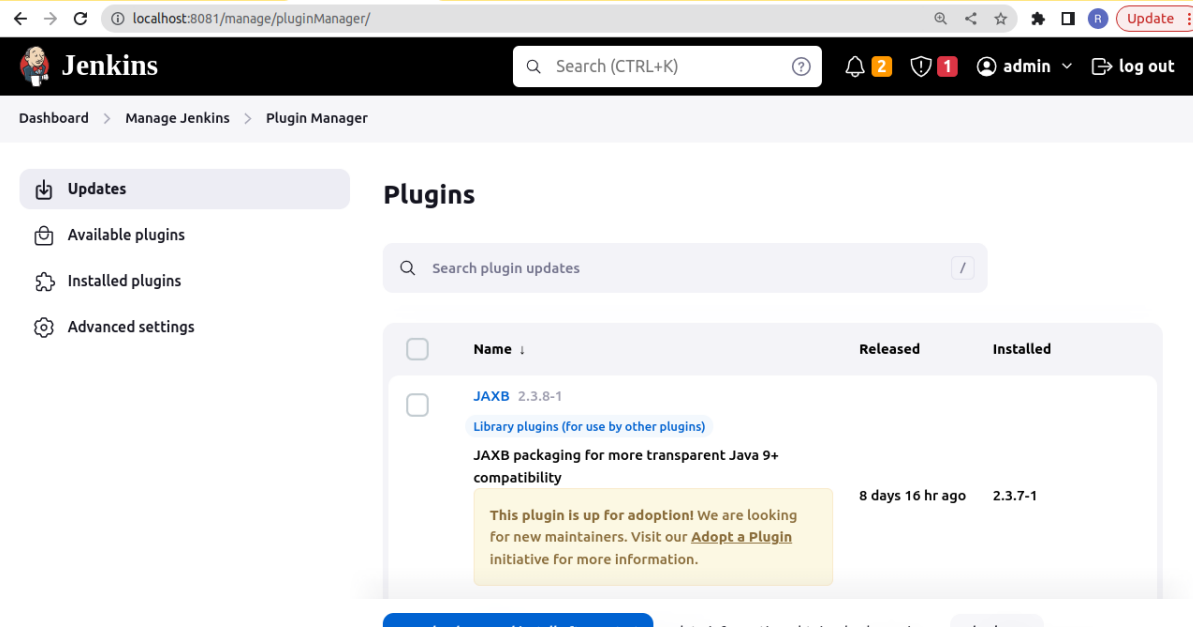
Configurar TomCat

Ahora vamos a configurar TomCat.

Para configurar TomCat tenemos que tener el Plugin de **Deploy to container**. Vamos a verlo brevemente cómo buscarlo.

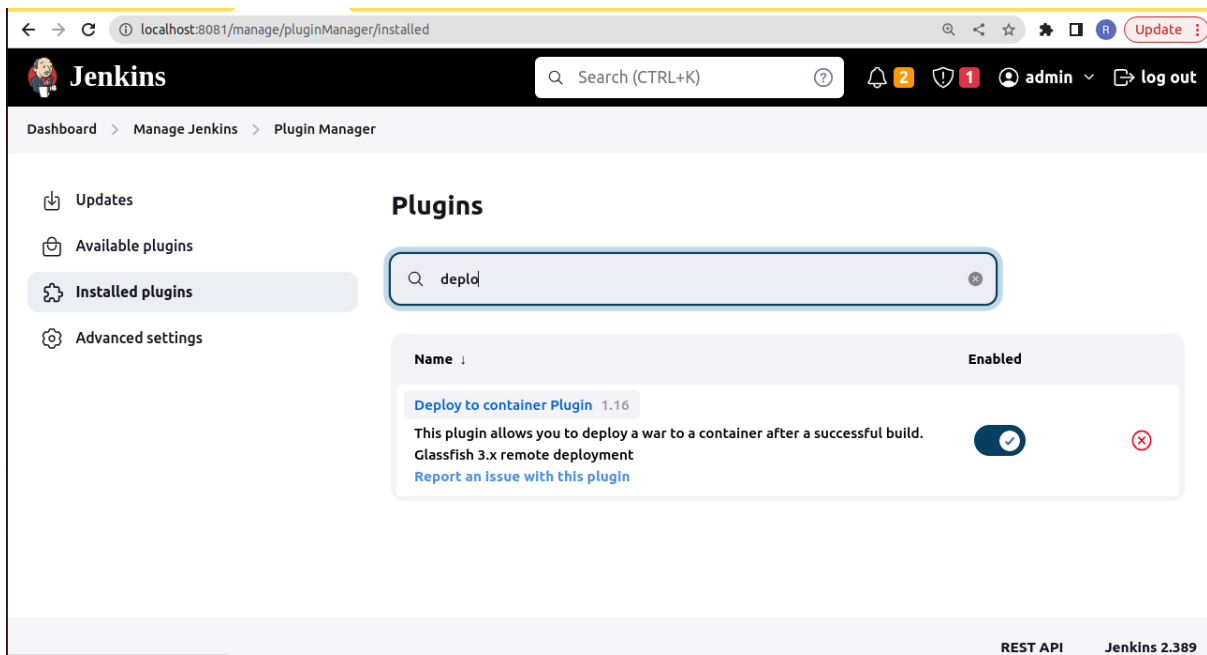


The screenshot shows the Jenkins 'Manage Jenkins' page. The left sidebar contains navigation links: 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (selected), and 'My Views'. Below these are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status'. The main content area is titled 'Manage Jenkins' and includes a search bar for settings. It features three informational banners: a blue banner about Jenkins 2.391, a yellow banner about security issues with built-in nodes, and another yellow banner about the Jenkins URL. At the bottom, there is a 'System Configuration' section.

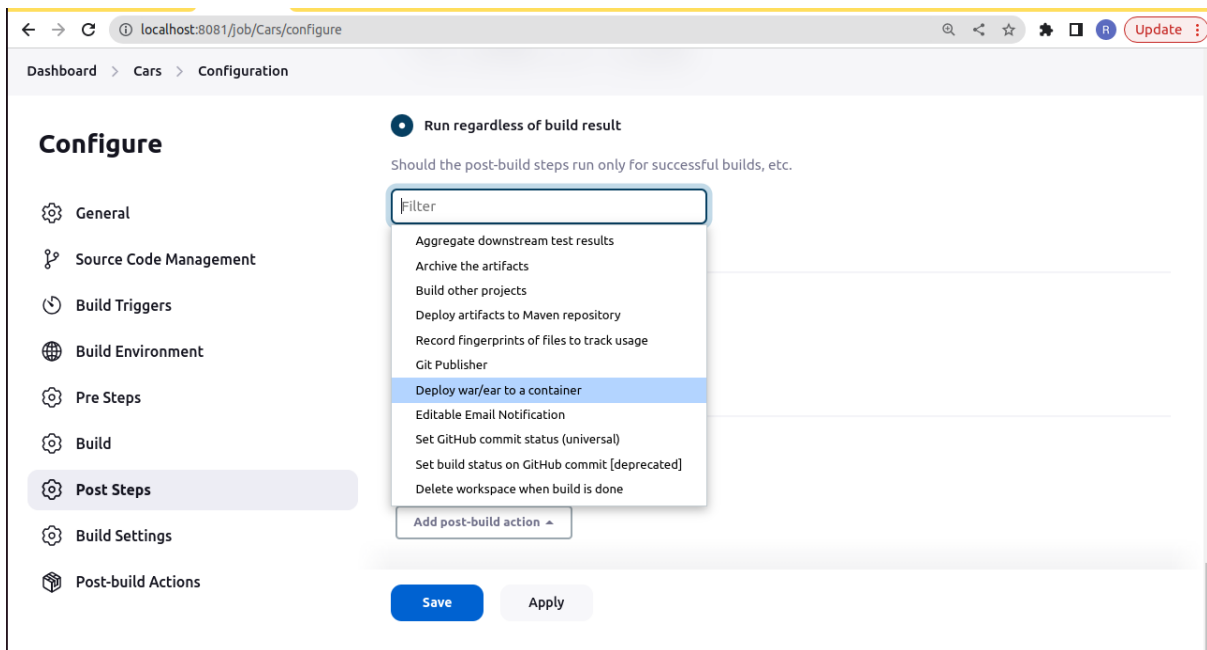


The screenshot shows the Jenkins 'Plugin Manager' page. The left sidebar has links for 'Updates' (selected), 'Available plugins', 'Installed plugins', and 'Advanced settings'. The main content area is titled 'Plugins' and includes a search bar for plugin updates. It displays a table of available plugins with columns for 'Name', 'Released', and 'Installed'. The first plugin listed is 'JAXB 2.3.8-1', which is a library plugin for Java 9+ compatibility. A yellow banner indicates that this plugin is up for adoption. At the bottom, there is a 'Download now and install after restart' button and a 'Check now' button.

Name	Released	Installed
JAXB 2.3.8-1 Library plugins (for use by other plugins) JAXB packaging for more transparent Java 9+ compatibility	8 days 16 hr ago	2.3.7-1

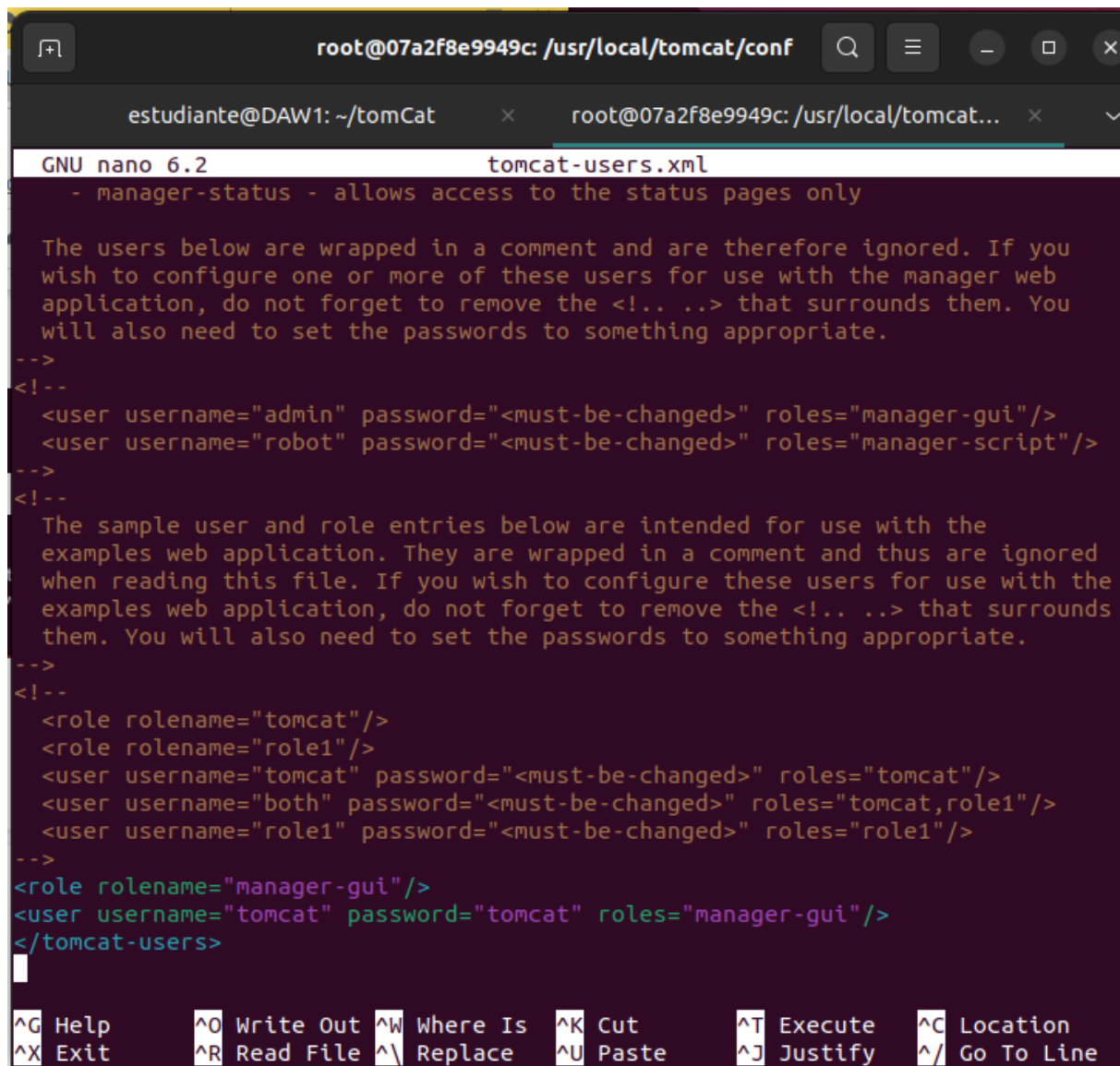


Una vez listo tenemos que ir otra vez a nuestro proyecto creado anteriormente e ir a **Post Steps > Run regardless of build result** y seleccionamos **Deploy war/ear to a container**.



```
estudiante@DAW1:~/tomCat$ docker start 1468622979ac
1468622979ac
estudiante@DAW1:~/tomCat$
```

```
root@07a2f8e9949c:/usr/local/tomcat/conf/Catalina# cd ..
root@07a2f8e9949c:/usr/local/tomcat/conf# ls
Catalina          context.xml        logging.properties  tomcat-users.xsd
catalina.policy   jaspic-providers.xml  server.xml           web.xml
catalina.properties  jaspic-providers.xsd  tomcat-users.xml
root@07a2f8e9949c:/usr/local/tomcat/conf# nano tomcat-users.xml
```



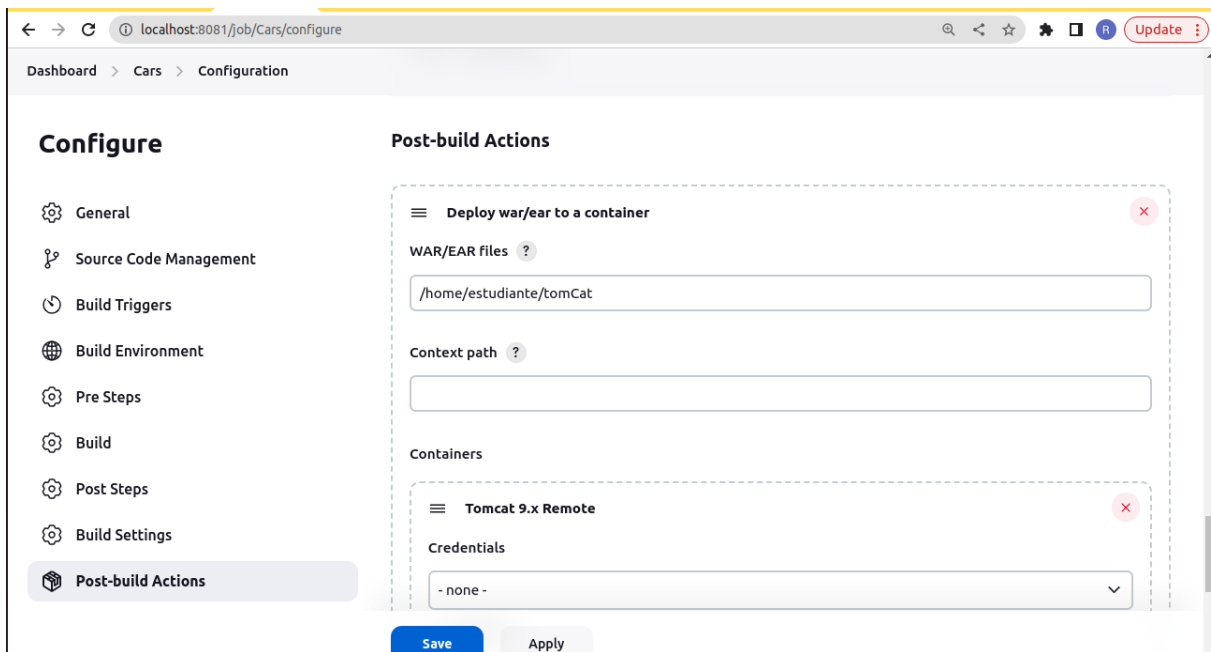
```
GNU nano 6.2 tomcat-users.xml
- manager-status - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!-- .. --> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- .. --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>
<user username="tomcat" password="tomcat" roles="manager-gui"/>
</tomcat-users>

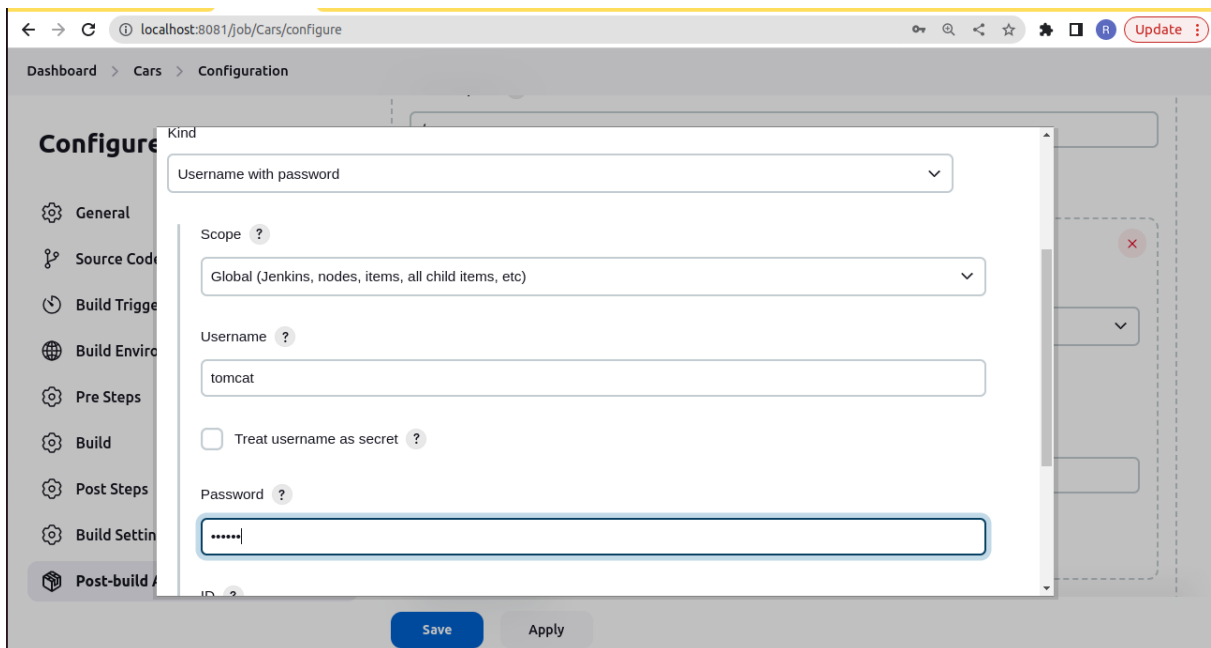
```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^/ Go To Line

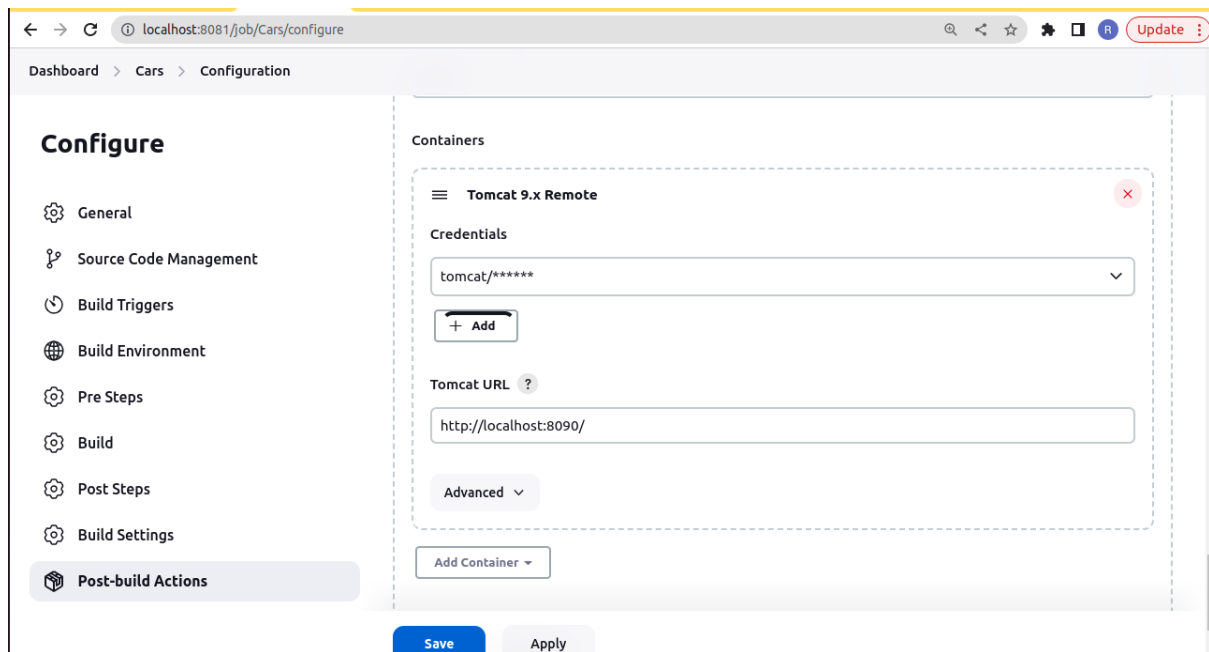
Después nos vamos a Post-build Actions > Deploy war/ear to a container > WAR/EAR files y escribimos la ruta donde tenemos instalado TomCat



Tenemos que bajar e poner el nombre y contraseña de TomCat



Lo último que tenemos que configurar en Tomcat es la url



Una vez guardado ya podremos usar nuestra aplicación.

Ya tenemos la aplicación funcionando perfectamente en nuestro Jenkins.



← → ↻

localhost:8081

🔍 ⏪ ⭐ ⚙️ 📄 🔄

Update

Dashboard >

+ New Item

👤 People

📅 Build History

🕒 Project Relationship

🔍 Check File Fingerprint

⚙️ Manage Jenkins

📌 My Views

All

+

S	W	Name ↓	Last Success	Last Failure	Last Duration
⊗	☁️	Cars	N/A	1 hr 2 min #9	13 sec

Icon: S M L Icon legend 📡 Atom feed for all 📡 Atom feed for failures 📡 Atom feed for just l

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

← → ↻

localhost:8081/job/Cars/lastFailedBuild/

🔍 ⏪ ⭐ ⚙️ 📄 🔄

Update

Dashboard > Cars > #9

📄 Status

</> Changes

📄 Console Output

📄 Edit Build Information

🗑️ Delete build '#9'

🔍 Git Build Data

🔄 Redeploy Artifacts

🔍 See Fingerprints

← Previous Build

⊗ Build #9 (Feb 15, 2023, 8:01:38 AM)

Keep this build forever

Add description

Started 1 day 0 hr ago

Took 13 sec

</> No changes.

Started by user [admin](#)

git

Revision: 542b5e273027f81ca8f1176b887003439f741324

Repository: <https://github.com/RomanMillan/cochesMillan2>

• refs/remotes/origin/main

Module Builds

⊗ [cochesMillan2](#) 2.9 sec

REST API Jenkins 2.389