

Introducción a Tomcat 10

En el siguiente documento se explicará como implantar un contenedor de servlets Tomcat versión 10 en un sistema Linux Debian.

Instalaciónsystemxctl

Para instalar tomcat10 desde repositorios:

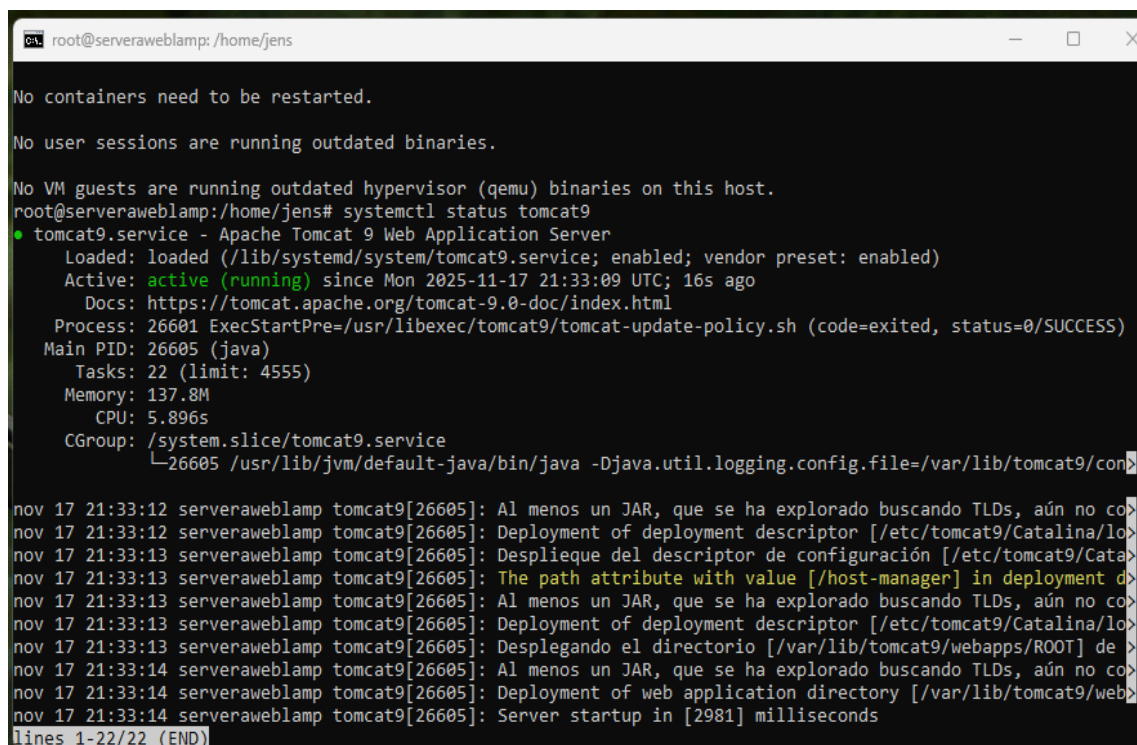
```
apt install tomcat10
```

Por dependencia va a instalar el paquete openjdk-17-jre-headless, que corresponde a una implementación de la JVM mínima para poder ejecutar nuestros programas java.

Desde este momento tendremos Tomcat ejecutándose y sirviendo en el puerto 8080.

Para gestionar el servicio tomcat:

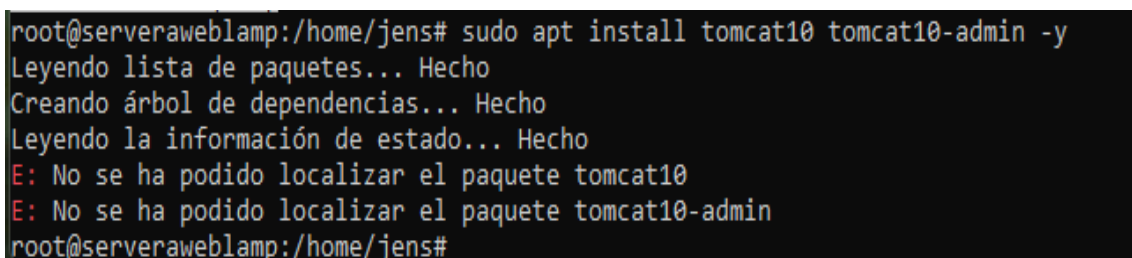
```
systemctl stop|start|restart|status tomcat10
```



```
root@serveraweblamp: /home/jens
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@serveraweblamp:/home/jens# systemctl status tomcat9
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-11-17 21:33:09 UTC; 16s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Process: 26601 ExecStartPre=/usr/libexec/tomcat9/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
   Main PID: 26605 (java)
     Tasks: 22 (limit: 4555)
    Memory: 137.8M
       CPU: 5.896s
   CGroup: /system.slice/tomcat9.service
           └─26605 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/con

nov 17 21:33:12 serveraweblamp tomcat9[26605]: Al menos un JAR, que se ha explorado buscando TLDs, aún no co
nov 17 21:33:12 serveraweblamp tomcat9[26605]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/lo
nov 17 21:33:13 serveraweblamp tomcat9[26605]: Despliegue del descriptor de configuración [/etc/tomcat9/Cata
nov 17 21:33:13 serveraweblamp tomcat9[26605]: The path attribute with value [/host-manager] in deployment d
nov 17 21:33:13 serveraweblamp tomcat9[26605]: Al menos un JAR, que se ha explorado buscando TLDs, aún no co
nov 17 21:33:13 serveraweblamp tomcat9[26605]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/lo
nov 17 21:33:13 serveraweblamp tomcat9[26605]: Desplegando el directorio [/var/lib/tomcat9/webapps/ROOT] de
nov 17 21:33:14 serveraweblamp tomcat9[26605]: Al menos un JAR, que se ha explorado buscando TLDs, aún no co
nov 17 21:33:14 serveraweblamp tomcat9[26605]: Deployment of web application directory [/var/lib/tomcat9/web
nov 17 21:33:14 serveraweblamp tomcat9[26605]: Server startup in [2981] milliseconds
lines 1-22/22 (END)
```

-He tenido que instalar el Tomcat9, por que el 10 no lo encontraba o no podía descargarlo. Adjunto captura del error.



```
root@serveraweblamp:/home/jens# sudo apt install tomcat10 tomcat10-admin -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete tomcat10
E: No se ha podido localizar el paquete tomcat10-admin
root@serveraweblamp:/home/jens#
```

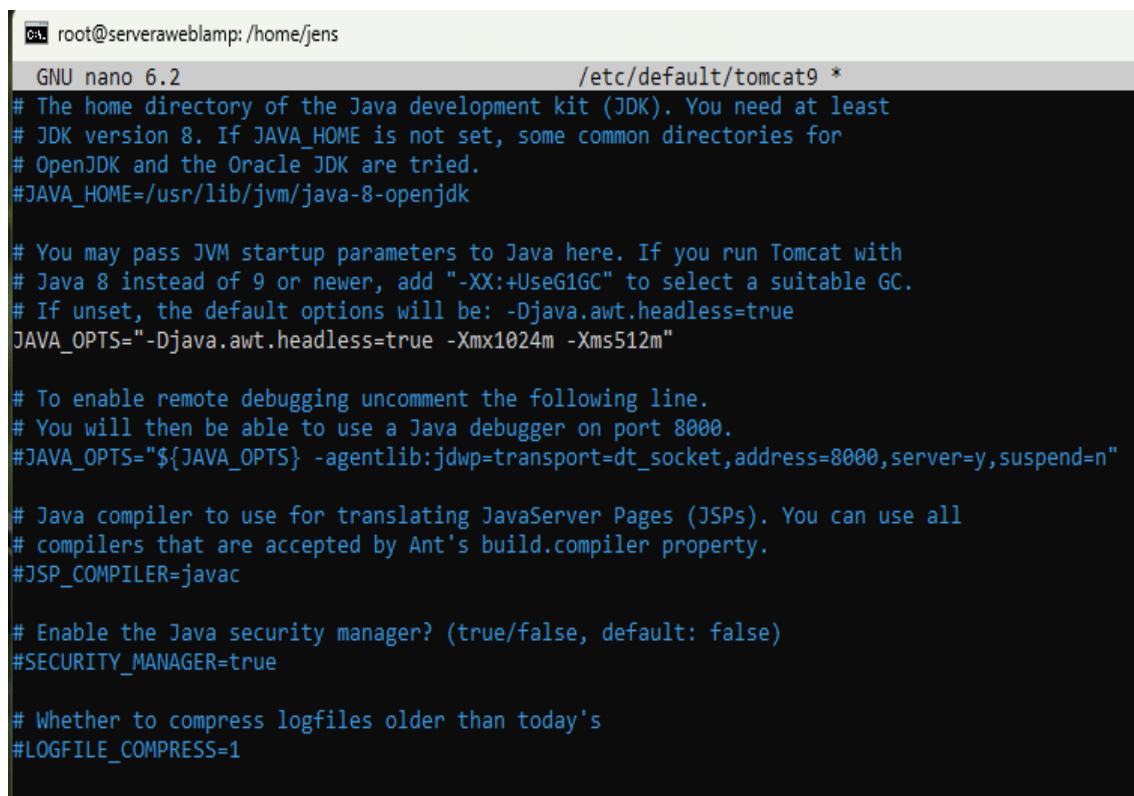
Opciones del servicio tomcat

Para modificar las opciones del servicio tomcat (que se guardan en la variable JAVA_OPTS) tenemos que indicarlo en el fichero /etc/default/tomcat10. Por defecto las opciones que tenemos configuradas son:

```
JAVA_OPTS="-Djava.awt.headless=true"
```

Para implantar algunas aplicaciones que necesitan mucha memoria RAM tenemos que añadir una opción indicando la cantidad de memoria que puede usar el servicio, por ejemplo para indicar 1Gb de memoria sería:

```
JAVA_OPTS="-Djava.awt.headless=true -Xmx1024m"
```



```
root@serveraweb: /home/jens
GNU nano 6.2 /etc/default/tomcat9 *
# The home directory of the Java development kit (JDK). You need at least
# JDK version 8. If JAVA_HOME is not set, some common directories for
# OpenJDK and the Oracle JDK are tried.
#JAVA_HOME=/usr/lib/jvm/java-8-openjdk

# You may pass JVM startup parameters to Java here. If you run Tomcat with
# Java 8 instead of 9 or newer, add "-XX:+UseG1GC" to select a suitable GC.
# If unset, the default options will be: -Djava.awt.headless=true
JAVA_OPTS="-Djava.awt.headless=true -Xmx1024m -Xms512m"

# To enable remote debugging uncomment the following line.
# You will then be able to use a Java debugger on port 8000.
#JAVA_OPTS="${JAVA_OPTS} -agentlib:jdwp=transport=dt_socket,address=8000,server=y,suspend=n"

# Java compiler to use for translating JavaServer Pages (JSPs). You can use all
# compilers that are accepted by Ant's build.compiler property.
#JSP_COMPILER=javac

# Enable the Java security manager? (true/false, default: false)
#SECURITY_MANAGER=true

# Whether to compress logfiles older than today's
#LOGFILE_COMPRESS=1
```

Despliegue de aplicaciones mediante la terminal

Implantar una aplicación desde la terminal, tampoco es tan difícil, ya que por defecto cualquier fichero .war que se copie o mueva dentro de l directorio /var/lib/tomcat10/webapps/ se desplegaría automáticamente y dependiendo de nuestra configuración se lanzaría o no.

En mucha documentación sobre tomcat se refiere a la variable de entorno \$CATALINA_HOME, en un debian donde hemos instalado tomcat con apt, el valor de esta variable será /var/lib/tomcat10.

Administración

Esta sección la iniciaremos utilizando una herramienta que nos proporciona la fundación Apache y que nos facilita el despliegue de aplicaciones y manejo del servidor, Tomcat-Manager. Para instalarlo:

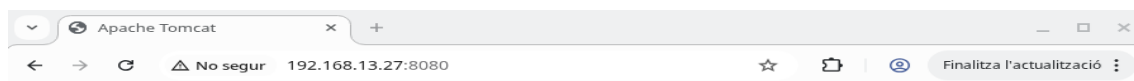
```
root@serverlemp:/etc/tomcat9# cat tomcat-users.xml
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">

  <!-- Roles para el Manager -->
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <role rolename="admin-gui"/>
  <role rolename="admin-script"/>

  <!-- Usuario administrador -->
  <user username="admin" password="admin123"
    roles="manager-gui,manager-script,manager-jmx,manager-status,admin-gui,admin-script"/>

  <!-- Usuario para solo ver el manager -->
  <user username="manager" password="manager123"
    roles="manager-gui,manager-status"/>

</tomcat-users>
root@serverlemp:/etc/tomcat9#
```



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.



apt install tomcat10-admin

Una vez instalado debemos crear un usuario con el rol manager para acceder a él. Añadimos una línea similar a la siguiente al fichero /etc/tomcat10/tomcat-users.xml:

```
<role rolename="manager-gui"/>
```

```
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Para acceder a la zona de administración:



Tomcat Web Application Manager

Message: OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status
-----------------------------------	-----------------------------------	------------------------------	-------------------------------

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy

WAR file to deploy

Select WAR file to upload No s'ha triat cap fixer
Deploy

Configuration

Re-read TLS configuration files



TLS host name (optional)

Despliegue de aplicaciones mediante la interfaz web

Utilizaremos la herramienta anterior para explicar cómo desplegar una aplicación, por ejemplo .war. Simplemente bajamos con el scroll hasta encontrar una sección llamada “WAR file to deploy”. Seleccionamos el fichero .war y le damos al botón “Deploy”.

Automáticamente se creará un nuevo elemento en la sección aplicaciones utilizando el mismo nombre que el fichero .war subido.

```
root@serveraweb1amp: ~/mi-aplicacion-web
root@serveraweb1amp:~/mi-aplicacion-web# ls
index.html  mi-aplicacion-web.war  WEB-INF
root@serveraweb1amp:~/mi-aplicacion-web#
```



Gestor de Aplicaciones Web de Tomcat

Mensaje: FALLO - Ya existe la aplicación en la trayectoria [/mi-aplicacion-web]

Gestor

[Gestionar Aplicaciones](#) | [Ayuda HTML de Gestor](#) | [Ayuda de Gestor](#) | [Estado de Servidor](#)

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
host-manager	Ninguno especificado	Tomcat Host Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos
mi-aplicacion-web	Ninguno especificado	Mi Aplicación Web	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 30 minutos

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Desde aquí podremos controlar la aplicación (Arrancarla, pararla, eliminarla,...)

Administración desde la terminal

Es hora de hablar de los ficheros de configuración.

server.xml

El más importante es /etc/tomcat10/server.xml, cuyo contenido define cómo está configurado nuestro servidor de aplicaciones java. Las secciones más importante de este fichero son:

- Componente **Server**: es el elemento principal del archivo server.xml y representa el servidor.

```
root@serveraweb1amp:~/mi-aplicacion-web# sudo grep -A10 "<Server" /etc/tomcat9/server.xml
<Server port="-1" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!-- APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
root@serveraweb1amp:~/mi-aplicacion-web#
```

- Componente **Service**: Un servicio es un grupo de conectores.

```
root@serveraweb1amp:~/mi-aplicacion-web# sudo grep -A15 "<Service" /etc/tomcat9/server.xml
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
    maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
  and responses are returned. Documentation at :
  Java HTTP Connector: /docs/config/http.html
  Java AJP Connector: /docs/config/ajp.html
  APR (HTTP/AJP) Connector: /docs/apr.html
  Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
  -->
root@serveraweb1amp:~/mi-aplicacion-web#
```

- o Subelemento **Connector**: El elemento Connector representa las conexiones (Puertos TCP) que serán abiertas por Tomcat al arranque, se definen diversos atributos los cuales dan más detalles acerca de la

```
root@serveraweb1amp: ~/mi-aplicacion-web
root@serveraweb1amp:~/mi-aplicacion-web# sudo grep -A5 "Connector port" /etc/tomcat9/server.xml
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
--
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="150" SSLEnabled="true">
  <SSLHostConfig>
    <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
      type="RSA" />
  </SSLHostConfig>
--
<Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
  maxThreads="150" SSLEnabled="true" >
  <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
  <SSLHostConfig>
    <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
      certificateFile="conf/localhost-rsa-cert.pem"
root@serveraweb1amp:~/mi-aplicacion-web#
```

conexión. Por defecto se define el puerto 8080. Nos encontramos comentados el conector para configurar HTTPS.

- o Subelemento **Engine**: Todas las peticiones que lleguen al servidor, se procesaran según la configuración de este elemento. Es similar a los VirtualHost de los servidores web.

```
root@serveraweblamp:~/mi-aplicacion-web# sudo grep -A10 "<Engine" /etc/tomcat9/server.xml
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine name="Catalina" defaultHost="localhost">

    <!--For clustering, please take a look at documentation at:
         /docs/cluster-howto.html (simple how to)
         /docs/config/cluster.html (reference documentation) -->
    <!--
    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
    -->

    <!-- Use the LockOutRealm to prevent attempts to guess user passwords
         via a brute-force attack -->
root@serveraweblamp:~/mi-aplicacion-web#
```

context.xml

Fichero de configuración específico de cada aplicación. Si alguna aplicación se despliega sin fichero context.xml, se aplicará la configuración del situado en /etc/tomcat10/context.xml. Su utilización es similar a la del fichero .htaccess de Apache.

```
root@serveraweblamp:~/mi-aplicacion-web# sudo cat /etc/tomcat9/context.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
</Context>
root@serveraweblamp:~/mi-aplicacion-web#
```

web.xml

Su ruta real es aplicacion/web-inf/web.xml, se trata de un descriptor de despliegue. Al igual que con el fichero context.xml, Tomcat posee un web.xml alojado en /etc/tomcat10/web.xml que se ejecuta antes del propio de cada aplicación. Con él se pueden activar y desactivar características como el compilador de JSP.

```
root@serveraweb1amp:~/mi-aplicacion-web# sudo cat /etc/tomcat9/web.xml | head -50
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the license. You may obtain a copy of the license at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the license for the specific language governing permissions and
limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">

  <!-- ===== Introduction ===== -->
  <!-- This document defines default values for *all* web applications -->
  <!-- loaded into this instance of Tomcat. As each application is -->
  <!-- deployed, this file is processed, followed by the -->
  <!-- "/WEB-INF/web.xml" deployment descriptor from your own -->
  <!-- applications. -->
  <!-- -->
  <!-- WARNING: Do not configure application-specific resources here! -->
  <!-- They should go in the "/WEB-INF/web.xml" file in your application. -->

  <!-- ===== Built In Servlet Definitions ===== -->

  <!-- The default servlet for all web applications, that serves static -->
  <!-- resources. It processes all requests that are not mapped to other -->
  <!-- servlets with servlet mappings (defined either here or in your own -->
  <!-- web.xml file). This servlet supports the following initialization -->
  <!-- parameters (default values are in square brackets): -->
  <!-- -->
  <!--   debug                Debugging detail level for messages logged -->
  <!--                       by this servlet. Useful values are 0, 1, and -->
  <!--                       11 where higher values mean more detail. [0] -->
  <!-- -->
  <!--   fileEncoding          Encoding to be used to read static resources -->
  <!--                       [platform default] -->
  <!-- -->
root@serveraweb1amp:~/mi-aplicacion-web#
```