

IES JACARANDÁ



**Despliegue de
Aplicaciones Web**

**Instalación de un
contenedor Tomcat**

Por: Chisela Marien Colás Gil

2 DAW

En este documento vamos a ver como crear un contenedor Tomcat que funciona como servidor web por sí mismo a partir de la [imagen oficial](#).

Dentro del directorio donde deseemos tener el servidor ejecutamos lo siguiente:

```
$ docker run -it -p 8888:8080 tomcat:9.0
```

```
estudiante@DAW1:~/Chisela/Despliegue web/tomcat$ docker run -it -p 8888:8080 tomcat:9.0
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /opt/java/openjdk
Using CLASSPATH:        /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
20-Jan-2023 07:13:13.486 INFO [main] org.apache.catalina.startup.VersionLoggerListener.lo
```

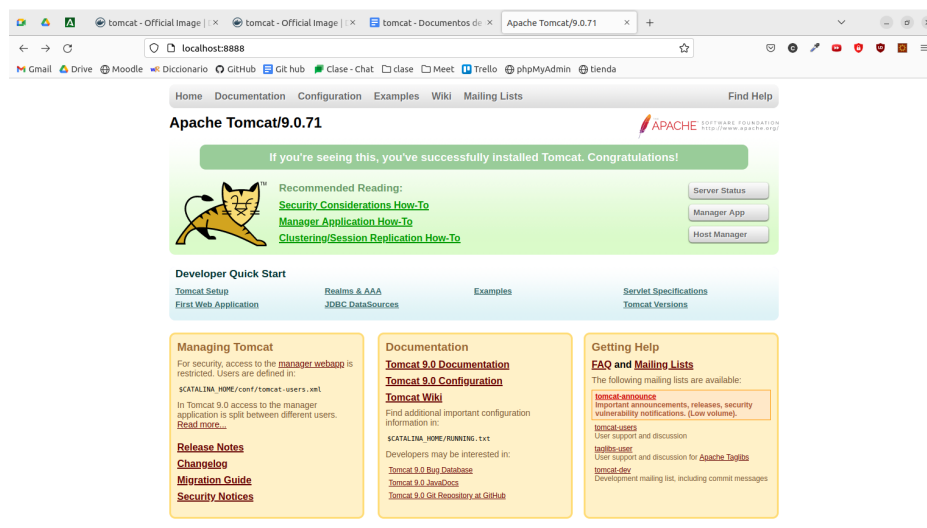
Ahora abrimos un nuevo terminal para ejecutar de forma interactiva el contenedor que acabamos de crear y modificar el fichero webapps con los siguientes comandos:

```
$ docker exec -it ID /bin/bash
```

```
$ cp -R webapps.dist/* webapps
```

```
estudiante@DAW1:~$ docker exec -it ee618bb535f2 /bin/bash
root@ee618bb535f2:/usr/local/tomcat# ls
bin  BUILDING.txt  conf  CONTRIBUTING.md  lib  LICENSE  logs  native-jni-lib  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  temp  webapps  webapps.dist  work
root@ee618bb535f2:/usr/local/tomcat# cp -R webapps.dist/* webapps
root@ee618bb535f2:/usr/local/tomcat# cd
bin/          conf/          lib/          logs/          NOTICE
RELEASE-NOTES  temp/          webapps.dist/
BUILDING.txt  CONTRIBUTING.md  LICENSE      native-jni-lib/  README.md
RUNNING.txt   webapps/       work/
```

Si ahora entramos a localhost:8888 (que es el puerto que hemos mapeado en el comando de creación) ya vemos la página funcionando. Sin embargo, si pulsamos en el botón “Manager App” de la derecha, nos llevará a una página de error ya que todavía no hemos configurado los permisos para esa parte.



Para configurar esta parte vamos a necesitar crear un fichero.xml así que necesitaremos el comando *nano*. Puede que lo tengamos ya instalado, pero si no es el caso simplemente ejecutamos *apt update* y después *apt install nano*. Nos movemos a la ruta */usr/local/tomcat/conf/Catalina/localhost* y creamos el archivo *manager.xml* con el comando *nano manager.xml*.

```
root@ee618bb535f2:/usr/local/tomcat/conf/Catalina/localhost# nano manager.xml
```

El contenido de este fichero es el siguiente:

```
<!-- 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 privileged="true" antiResourceLocking="false"
docBase="${catalina.home}/webapps/manager"> <!-- Default set of monitored
resources. If one of these changes, the --> <!-- web application will be reloaded. -->
<WatchedResource>WEB-INF/web.xml</WatchedResource>
<WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
<WatchedResource>${catalina.base}/conf/web.xml</WatchedResource> <Valve
className="org.apache.catalina.valves.RemoteAddrValve" allow="^.*$"/> </Context>
<!-- Uncomment this to disable session persistence across Tomcat restarts --> <!--
<Manager pathname="" /> -->
```

Por último, tenemos que modificar las últimas líneas del fichero *tomcat-users.xml* (en el directorio *conf*) para añadir los roles e usuarios necesarios antes de la etiqueta *</tomcat-users>*, que marca el final del documento. Ejecutamos el comando *nano tomcat-users.xml* y añadimos lo siguiente:

```
root@ee618bb535f2:/usr/local/tomcat/conf# nano tomcat-users.xml
```

...

```
<role rolename="tomcat"/>
<user username="tomcat" password="tomcat" roles="manager-gui"/>
</tomcat-users>
```

Una vez completado este paso, volvemos a nuestra página en localhost:8888, accedemos a “Manager App”, introducimos el usuario(tomcat) y la contraseña (tomcat) y ya tenemos el servidor listo.

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	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	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