

Instalación de Apache Tomcat como servidor de Aplicaciones:

Deberemos instalar Apache Tomcat en una de las maquinas apache configuradas con debian previamente, para ello deberemos tener instalado java y usaremos el siguiente comando:

“sudo apt install openjdk-17-jdk -y”

```
a24gadilu@debian:~$ sudo apt install openjdk-17-jdk -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  nginx-common
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  adwaita-icon-theme at-spi2-common at-spi2-core ca-certificates-java dbus-user-session dconf-gsettings-backend
  dconf-service fontconfig fontconfig-config fonts-dejavu-core fonts-dejavu-extra gsettings-desktop-schemas
  gtk-update-icon-cache hicolor-icon-theme java-common libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni
  libatk1.0-0 libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2
  libcups2 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libdrm2 libfontconfig1 libfontenc1 libfribidi0 libgail-common libgail18 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
  libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3
  libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg62-turbo liblcms2-2
  liblerc4 libllvm15 libnspr4 libnss-systemd libnss3 libpam-systemd libpango-1.0-0 libpangocairo-1.0-0
  libpangoft2-1.0-0 libpciaccess0 libpixmap-1-0 libpthread-stubs0-dev librsvg2-2 librsvg2-common libsensors-config
  libsensors5 libsm-dev libsm6 libthai-data libthai0 libtiff6 libwebp7 libx11-dev libx11-xcb1 libxau-dev libxaw7
```

Instalamos tomcat10:

“sudo apt install tomcat10 -y”

```
a24gadilu@debian:~$ sudo apt install tomcat10 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  nginx-common
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libeclipse-jdt-core-java libtcnative-1 libtomcat10-java tomcat10-common
Paquetes sugeridos:
  tomcat10-admin tomcat10-docs tomcat10-examples tomcat10-user
Se instalarán los siguientes paquetes NUEVOS:
```

Creamos usuario y lo añadimos a un grupo tomcat:

“sudo groupadd tomcat10”

```
a24gadilu@debian:~$ sudo groupadd tomcat10
```

“sudo useradd -s /bin/false -g tomcat10 -d /etc/tomcat10 tomcat10”

```
a24gadilu@debian:~$ sudo useradd -s /bin/false -g tomcat10 -d /etc/tomcat10 tomcat10
```

Arrancamos tomcat10 y vemos que esta activado:

“sudo systemctl start tomcat10”

“sudo systemctl status tomcat10”

```

a24gadilu@debian:~$ sudo systemctl start tomcat10
a24gadilu@debian:~$ sudo systemctl status tomcat10
● tomcat10.service - Apache Tomcat 10 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat10.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-05 10:27:52 CET; 1min 12s ago
     Docs: https://tomcat.apache.org/tomcat-10.0-doc/index.html
   Process: 4799 ExecStartPre=/usr/libexec/tomcat10/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
   Main PID: 4803 (java)
    Tasks: 37 (limit: 2306)
   Memory: 100.6M
      CPU: 3.359s
   CGroup: /system.slice/tomcat10.service
           └─4803 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.util.logging.config.file=/var/lib/tomcat10/conf/

nov 05 10:27:52 debian tomcat10[4803]: OpenSSL inicializado correctamente [OpenSSL 3.0.14 4 Jun 2024]
nov 05 10:27:52 debian tomcat10[4803]: Inicializando el manejador de protocolo ["http-nio-8080"]
nov 05 10:27:52 debian tomcat10[4803]: Server initialization in [448] milliseconds
nov 05 10:27:52 debian tomcat10[4803]: Arrancando servicio [Catalina]
nov 05 10:27:52 debian tomcat10[4803]: Starting Servlet engine: [Apache Tomcat/10.1.6 (Debian)]
nov 05 10:27:52 debian tomcat10[4803]: Desplegando el directorio [/var/lib/tomcat10/webapps/ROOT] de la aplicación web
nov 05 10:27:53 debian tomcat10[4803]: Al menos un JAR, que se ha explorado buscando TLDs, aún no contenía TLDs. Activas
nov 05 10:27:53 debian tomcat10[4803]: Deployment of web application directory [/var/lib/tomcat10/webapps/ROOT] has fin
nov 05 10:27:53 debian tomcat10[4803]: Starting ProtocolHandler ["http-nio-8080"]
nov 05 10:27:53 debian tomcat10[4803]: Server startup in [1003] milliseconds
lines 1-22/22 (END)

```

Ahora deberemos añadir el usuario al archivo de configuración xml de tomcat, este archivo se encuentra en la siguiente ruta: /etc/tomcat10/tomcat10-users.xml

```

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

<user username="a24gadilu" password="3367G103@" roles="admin,admin-gui,manager,manager-gui"/>
</tomcat-users>

```

Ahora accederemos a la dirección ip de nuestra maquina con tomcat instalado para ver si funciona correctamente. Debemos poner la dirección ip (192.168.116.15) junto con el puerto que en este caso es el 8080:

```

← → ↻ 🌐 http://192.168.116.15:8080

```

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/tomcat10/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/tomcat10 and CATALINA_BASE in /var/lib/tomcat10, following the rules from /usr/share/doc/tomcat10-common/RUNNING.txt.gz.

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

tomcat10-docs: This package installs a web application that allows to browse the Tomcat 10 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat10-examples: This package installs a web application that allows to access the Tomcat 10 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat10-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/tomcat10/tomcat-users.xml.

Ahora deberemos descargar el web manager de tomcat10 con el siguiente comando:

“sudo apt install tomcat10-admin”

```
a24gadilu@debian:~$ sudo apt install tomcat10-admin
[sudo] contraseña para a24gadilu:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 tomcat10-admin
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 71,3 kB de archivos.
Se utilizarán 327 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 tomcat10-admin all 10.1.6-1+deb12u2 [71,3 kB]
Descargados 71,3 kB en 0s (165 kB/s)
Seleccionando el paquete tomcat10-admin previamente no seleccionado.
(Leyendo la base de datos ... 41276 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../tomcat10-admin_10.1.6-1+deb12u2_all.deb ...
Desempaquetando tomcat10-admin (10.1.6-1+deb12u2) ...
Configurando tomcat10-admin (10.1.6-1+deb12u2) ...
a24gadilu@debian:~$
```

Ahora accederemos a la ruta para ver si nos pide usuario y contraseña:

Ruta: <http://192.168.116.15:8080/manager/html>

192.168.116.15:8080/manager/html

Iniciar sesión

<http://192.168.116.15:8080>



Tu conexión con este sitio web no es privada

Nombre de usuario

Contraseña

Metemos usuario y contraseña para entrar y como podemos observar en la imagen ya estamos dentro de Apache Tomcat:

← → ↻ No es seguro 192.168.116.15:8080/manager/html ☆ 🔴



Gestor de Aplicaciones Web de Tomcat

Mensaje: OK

Gestor

Listar 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	0	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

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

Para entrar por FQDN y proxy inverso desde nuestro servidor nginx deberemos cambiar la configuración de nuestro archivo de configuración de nginx en este caso el archivo se llama a24gadilu:

Deberemos añadir la dirección ip interna de la maquina que tiene tomcat instalado.

```
GNU nano 7.2 /etc/nginx
server {
    listen 80;
    server_name www.a24gadilu.local;

    location / {
        proxy_pass http://192.168.201.10:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Una vez cambiada la configuración deberemos reiniciar el servicio de nginx y entramos a la url de nuestro servidor, en este caso es <http://www.a24gadilu.local>.



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/tomcat10/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/tomcat10` and `CATALINA_BASE` in `/var/lib/tomcat10`, following the rules from `/usr/share/doc/tomcat10-common/RUNNING.txt.gz`.

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

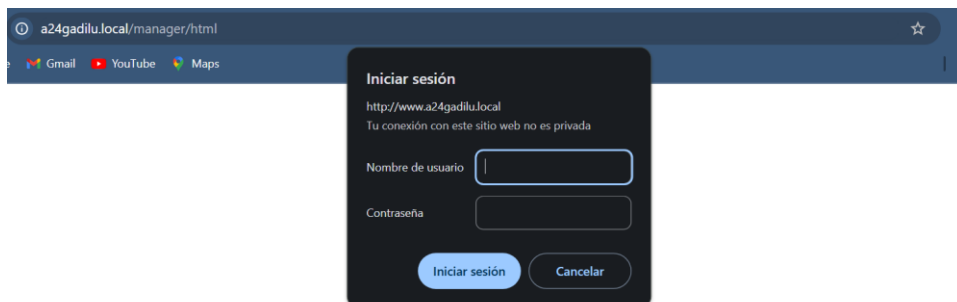
tomcat10-docs: This package installs a web application that allows to browse the Tomcat 10 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat10-examples: This package installs a web application that allows to access the Tomcat 10 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat10-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/tomcat10/tomcat-users.xml`.

Accedemos al manager de tomcat y nos pedirá la contraseña y usuario:

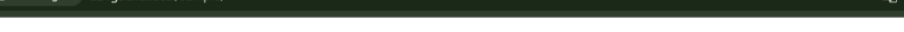


Archivo WAR a desplegar

Seleccione archivo WAR a cargar sample.war

Desplegar

Gestor						
Listar 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	0	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	
/sample	Ninguno especificado	Hello, World Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos	



The screenshot shows a web browser window with the address bar displaying 'a24gadilu.local/sample/'. The page title is 'Sample "Hello, World" Application'. On the left, there is a yellow cat icon. The main text states: 'This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.' Below this, it says 'To prove that they work, you can execute either of the following links:' followed by two bullet points: '• To a [JSP page.](#)' and '• To a [servlet.](#)'

[illegible]

Deberemos cambiar la configuración del servidor nginx donde deberemos redireccionar

el puerto para que escuche en el 443 y donde coja el certificado y la clave que hemos creado previamente:

```
GNU nano 7.2 sites-available/a24gadilu
server {
    listen 80;
    server_name www.a24gadilu.local;

    location / {
        return 301 https://$host$request_uri;
    }
}

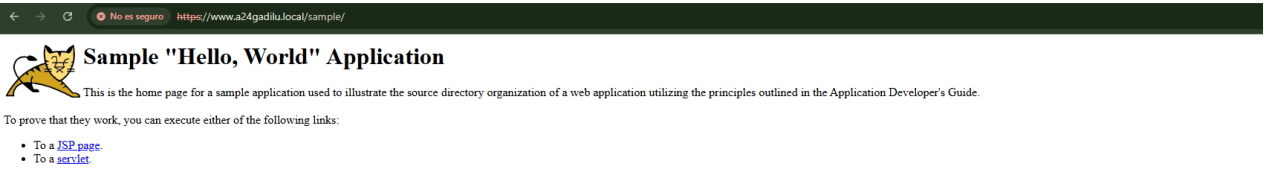
server {
    listen 443 ssl;
    server_name www.a24gadilu.local;

    ssl_certificate /etc/nginx/ssl/a24gadilu.crt;
    ssl_certificate_key /etc/nginx/ssl/a24gadilu.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    location / {
        proxy_pass http://192.168.201.10:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
    }
}
```

Guardamos configuración y reiniciamos el servicio de nginx. Accedemos a nuestra url del servidor (<https://www.a24gadilu.local/sample>) y podemos ver que ya hemos entrado por certificado SSL. Nos saldrá que no es seguro debido a que lo hemos creado nosotros.



Aquí podemos ver las especificaciones del certificado:



De esta manera ya tendremos nuestro servidor nginx que apunta a Apache tomcat a través del proxy inverso y podemos acceder a él a través de FQDN y por certificado SSL.