Automatización de la recogida de firmas para iniciativas ciudadanas

Anexo VI: Manual de despliegue

VNiVERSiDAD D SALAMANCA

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Julio de 2016

Autor

Aythami Estévez Olivas

Tutor

Rodrigo Santamaría Vicente

Índice de contenidos

Índice de figuras

1. Introducción

Este anexo forma parte de la documentación técnica del proyecto "Automatización de la recogida de firmas para iniciativas ciudadanas", conocido de una forma más informal como **Demos**.

En él se recogen las instrucciones para el despliegue del proyecto en el servidor. Respecto al cliente Android basta con instalarlo en el dispositivo como una aplicación más.

Las instrucciones aquí dadas están pensadas para la instalación y despliegue del servidor en un entorno **Debian**, aunque en principio sería posible desplegarlo en otros entornos siempre y cuando puedan ejecutar un servidor de aplicaciones **Apache Tomcat**.

2. Paquetes básicos

Antes comenzar con la instalación y configuración del servidor se requieren ciertos paquetes básicos instalados. Estos son:

- ***build-esential****:* necesario para la compilación de código en entornos Linux.
- ***JDK y JRE*****:** paquete de desarrollo y máquina virtual de Java, se recomienda la versión oficial de Oracle en lugar de las versiones *Open-JDK*. En concreto para este proyecto se ha empleado la versión 1.8.0_91. Para instrucciones de instalación vaya a [1].
- ***ant*[2]*:*** necesario para la compilación de ciertos paquetes.
- ***cmake:*** necesario por el mismo motivo que el anterior.

Si no se indica lo contrario se recomienda instalar los paquetes usando la herramienta de gestión de paquetes propia del SO, en el caso de Debian se ha utilizado **apt-get.**

[1] http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-debian.html
[2] https://ant.apache.org/

3. Apache Tomcat

Lo mejor es instalarlo via apt-get, ya que al hacerlo se instala como un servicio más del sistema (se ejecuta automáticamente en el arranque, tiene usuario propio). En concreto se ha utilizado la versión 8.0.14, pero cualquier versión 8.x seria perfectamente válida. Es posible que también se pueda hacer con versiones distintas pero su uso no está probado.

Al instalarlo via apt-get se instalan con él las siguientes dependencias:

- ***authbind:*** Esta utilidad será utilizada más adelante.
- ***libcommons-pool-java***
- ***libcommons-dbcp-java***
- ***libecj-java***
- ***libtomcat8-java***
- ***tomcat8-common***
- ***tomcat8***
- ***tomcat8-admin***

La otra opción para ejecutarlo es descargarlo y lanzarlo manualmente ejecutando el script *./catalina.sh run* desde el directorio */bin* de Tomcat.

4. OpenCV

Es necesario instalar OpenCV para la detección de DNIs. Hay que descargar el paquete que se encuentra en la página oficial y que contiene todos los fuentes, posteriormente hay que compilarlo con ciertas opciones para instalar la versión de Java. En concreto se ha utilizado la versión 3.1.0 de esta librería. Para instrucciones detalladas de instalación en distintos entornos vaya a [3].

Importante: Tras haberlo instalado es necesario acceder a *OPENCV_INSTALLATION_DIRECTORY/build/lib/* y copiar o mover el archivo *libopencv_java310.so* a las rutas de inclusión de librerías de java (java.library.path), en caso contrario Tomcat no será capaz de cargar la librería y el servidor fallara al intentar detectar un DNI, para consultar esas rutas usar el comando *java -XshowSettings:properties*.

[3] http://opencv-java-tutorials.readthedocs.io/en/latest/01-installing-opencv-for-java.html

5. Tesseract

Existen dos opciones para la instalación de este motor de OCR:

- Instalación via apt-get: es necesario instalar los paquetes ***tesseract-ocr*** y ***tesseract-ocr-spa***.
- Descargar y compilar el código fuente: para ello vaya a [4].

Para elegir una de las opciones introduzca el comando *locale* en su ordenador. Si no le apareces todas las líneas igualadas a *"C"* se recomienda que descargue el código fuente y lo compile por su cuenta ya que se han encontrado problemas con la codificación de caracteres que utiliza Tesseract instalando los binarios únicamente.

[4] https://github.com/tesseract-ocr/tesseract/wiki/Compiling

6. Configuración de Tomcat para funcionar con HTTPS

Debe conseguir un certificado X509 para firmar el certificado del servidor, esto se puede hacer recurriendo a una CA oficial o creando un certificado auto firmado usted mismo.

Una vez hecho eso siga las instrucciones que aparecen en [5]. Debe crear una ***Certificate Signing Request (CSR)*** y firmarla con su certificado. Tras esto puede importarlo al Keystore de Tomcat.

Ahora es necesario configurar el servidor para que establezca un conector seguro, esto se ha hecho añadiendo el siguiente fragmento al fichero *server.xml* que se encuentra en */etc/tomcat8/*.

```
<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    keystoreFile="/etc/tomcat8/tomcat_keystore.jks" keystorePass="canario010694"
    keyAlias="tomcat" clientAuth="false" sslProtocol="TLS"
    sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2" useServerCipherSuitesOrder="true"
    ciphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
        TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,
        TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,
        TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,
        SSL_RSA_WITH_3DES_EDE_CBC_SHA"/>
```

ILUSTRACIÓN 1: CONFIGURACIÓN DE CONECTOR SEGURO TOMCAT

Tras esto debe permitir a Tomcat ejecutarse en el puerto 443. Véase la herramienta ***authbind*** en [6].

[5] https://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html
[6] https://debian-administration.org/article/386/Running_network_services_as_a_non-root_user

7. Consideraciones finales

Para finalizar debe definir una contraseña para proteger el Keystore que utilizara el proyecto. Para ello es necesario crear el archivo ***.DemosKey*** en */etc/tomcat8/* con la contraseña que desee utilizar (sin añadir ningún espacio o línea en blanco), cambiar el propietario de ese archivo al usuario tomcat8 y darle solo permisos de lectura a él.

Como último paso cree el directorio ***/demos*** haciendo propietario a tomcat8 y dándole permisos de lectura, escritura y ejecución solo a él.