

8.18. Seguridad SSL

- Habilita SSL en el *Host* por defecto de *Tomcat* (*localhost*).
- Configura la aplicación **curso** para que solo permita conexiones https.

1. Crear un almacén de claves con un certificado SSL.

- 1.1. Inicia sesión en **ServidorLinuxXX** con un usuario con privilegios de administración.
- 1.2. Utiliza la herramienta `keytool` para crear un certificado autofirmado y almacenarlo en un almacén de claves JKS.

```
sudo keytool -genkey -alias tomcat -keyalg RSA \  
-keystore /var/lib/tomcat7/daw01keystore
```

Introduce los datos del certificado, Figura 8.129

```

alumno@ServidorLinux01:/var/lib/tomcat7$ sudo keytool -genkey -alias tomcat -key
alg RSA -keystore /var/lib/tomcat7/daw01keystore
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: servidorlinux01.daw01.net
¿Cuál es el nombre de su unidad de organización?
[Unknown]: tomcat
¿Cuál es el nombre de su organización?
[Unknown]: daw01.net
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Madrid
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Madrid
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ES
¿Es correcto CN=servidorlinux01.daw01.net, OU=tomcat, O=daw01.net, L=Madrid, ST=
Madrid, C=ES?
[no]: si

Introduzca la contraseña de clave para <tomcat>
(INTRO si es la misma contraseña que la del almacén de claves):
alumno@ServidorLinux01:/var/lib/tomcat7$

```

Figura 8.129: Creación un certificado autofirmado dentro de un almacén de claves

Durante la generación del certificado se tienen que introducir dos claves, la del almacén de claves (**daw01keystore**) y la de las claves asociadas a alias (**tomcat**) creado. En ambos casos pondremos como clave **despliegue**.

2. Configurar un conector SSL en *Tomcat*.

- 2.1. En **ServidorLinuxXX** edita el fichero `/var/lib/tomcat7/conf/server.xml/` y realiza la configuración que se muestra en la Figura 8.130.

```

<!-- Define a SSL HTTP/1.1 Connector on port 8443
This connector uses the JSSE configuration, when using APR, the
connector should be using the OpenSSL style configuration
described in the APR documentation -->

<Connector
  port="8443"
  protocol="HTTP/1.1"
  SSLEnabled="true"
  maxThreads="150"
  scheme="https" secure="true"
  clientAuth="false"
  sslProtocol="TLS"
  keystoreFile="/var/lib/tomcat7/daw01keystore" keystorePass="despliegue"
  keyAlias="tomcat" keyPass="despliegue"

```

Figura 8.130: Fichero `/var/lib/tomcat7/conf/server.xml/`

- 2.2. Reinicia *Tomcat*.
- 2.3. Comprueba que el servidor está iniciado y escuchando en los puerto 8080/TCP y 8443/TCP.

```

ps -ef | grep tomcat
netstat -ltn

```

- 2.4. Desde **DesarrolloW7XX** abre el navegador y establece una conexión a `https://servidorlinux01.daw01.net:8443/curso`, Figuras 8.131, 8.132 y 8.133.

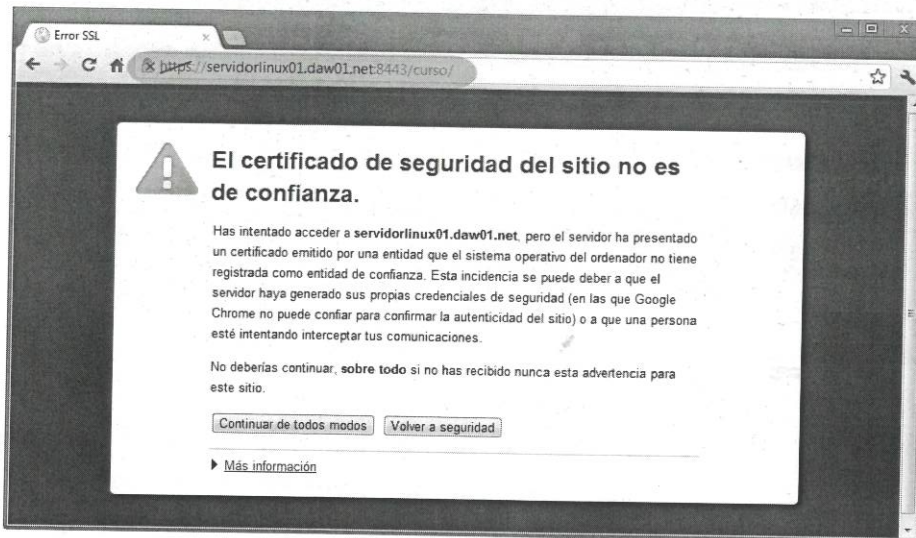


Figura 8.131: Conexión https

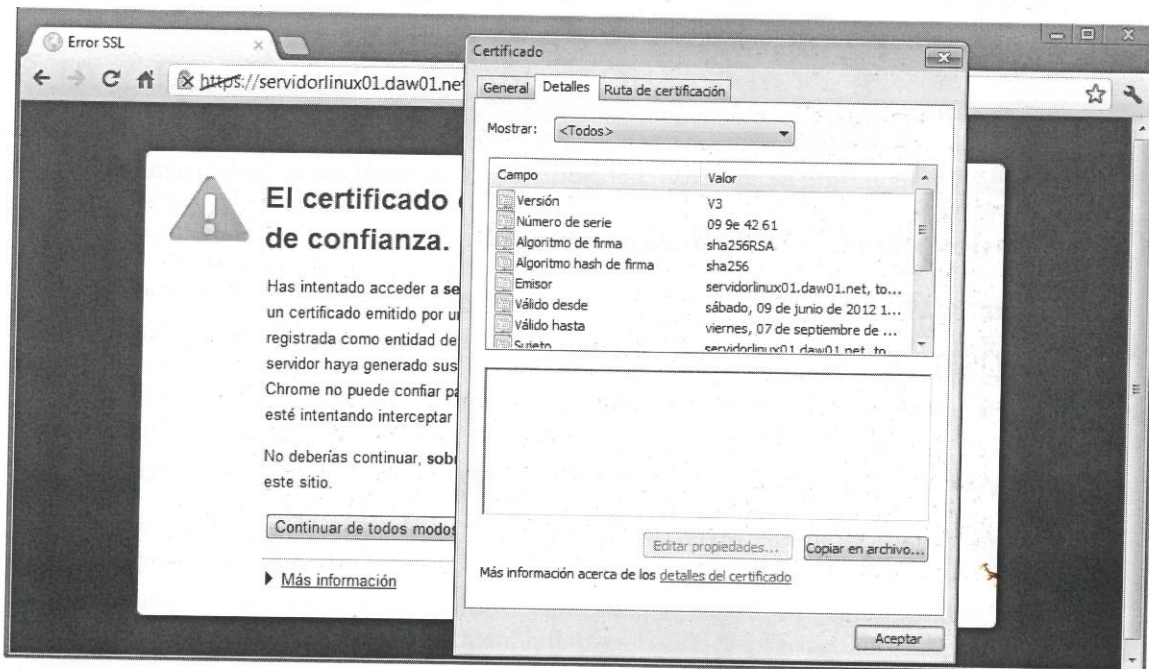


Figura 8.132: Conexión https

3. Configurar la aplicación curso para que solo permita conexiones https.

- 3.1. Desde **DesarrolloW7XX** abre el navegador y establece una conexión a `http://servidorlinux01.daw01.net:8080/curso` y observa que permite conexiones http, Figura 8.134.

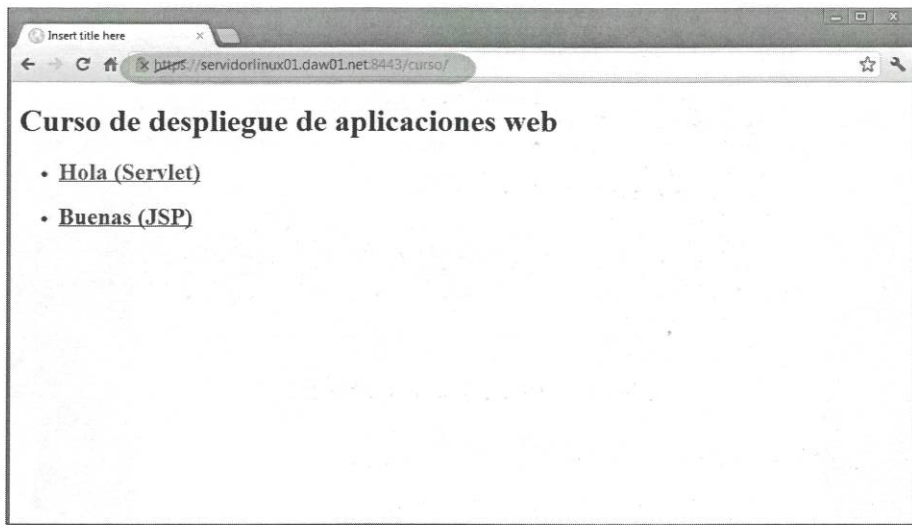


Figura 8.133: Conexión https

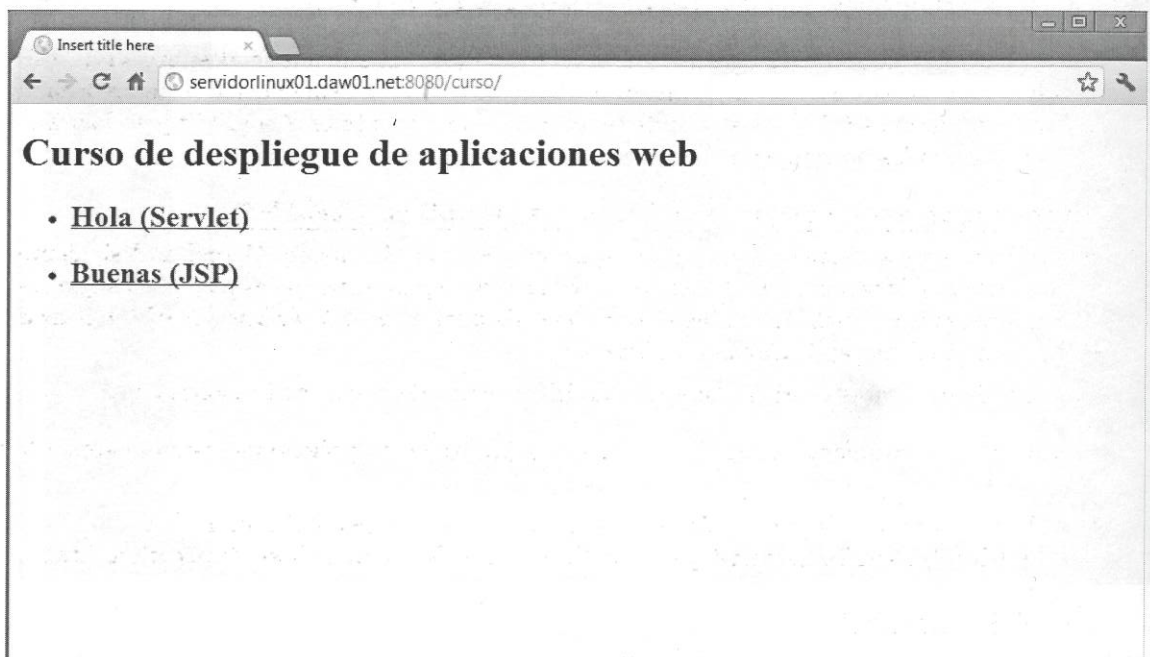


Figura 8.134: Conexión http

- 3.2. Edita el descriptor de despliegue de la aplicación **curso** y realiza la siguiente configuración, Figura 8.135.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>MemoryRealm</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>curso</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Figura 8.135: Fichero **web.xml**

- 3.3. Elimina la aplicación **curso** del servidor *Tomcat*.
- 3.4. Genera el fichero **curso.war** con los cambios.
- 3.5. Despliega el nuevo fichero **curso.war**.
- 3.6. Desde **DesarrolloW7XX** abre el navegador y establece una conexión a <http://servidorlinux01.daw01.net:8443/curso> y observa la petición se redirige a <https://servidorlinux01.daw01.net:8443/curso>.

8.19. Despliegue en la nube (OpenShift) - 3

En esta práctica vamos a desplegar la aplicación **curso** (la primera versión desarrollada) utilizando las **herramientas en línea de comandos** de *OpenShift*. Se presupone que se han realizado previamente las prácticas 6.8 y 6.9 en donde se ha creado una cuenta en *OpenShift* y se han instalado y configurado las herramientas necesarias.

1. Introducción al despliegue de aplicaciones JavaEE en *OpenShift*

OpenShift permite desplegar y ejecutar aplicaciones Java fácilmente. Consulta el siguiente enlace <https://www.openshift.com/developers/java> para conocer las tecnologías que soporta. Observa que está disponible el *cartridge Tomcat 6 and 7 (JBoss EWS 1.0 and 2.0)* para desplegar aplicaciones en *Tomcat*.

Los siguientes enlaces contiene información útil de cómo desplegar aplicaciones Java:

- <https://www.openshift.com/blogs/run-your-java-tomcat-application-for-free-on-openshifts-paas>
- https://access.redhat.com/site/documentation/en-US/OpenShift_Online/2.0/html/User_Guide/sect-OpenShift-User_Guide-Deploying_JBoss_Applications.html

2. Crear la aplicación

- 2.1. Inicia sesión en **DesarrolloW7XX** con un usuario con privilegios de administrador.
- 2.2. Ejecuta el siguiente comando para mostrar los **cartridges** disponibles, Figura 8.136.

```
rhc cartridge list
```

php-5.3	PHP 5.3	web
python-2.6	Python 2.6	web
python-2.7	Python 2.7	web
python-3.3	Python 3.3	web
ruby-1.8	Ruby 1.8	web
ruby-1.9	Ruby 1.9	web
jbosses-1.0	Tomcat 6 <JBoss EWS 1.0>	web
jbosses-2.0	Tomcat 7 <JBoss EWS 2.0>	web
zend-5.6	Zend Server 5.6	web
diy-0.1	Do-It-Yourself 0.1	web
logen-mms-agent-0.1	logen Mongo Monitoring Service Agent	addon
cron-1.4	Cron 1.4	addon
jenkins-client-1.4	Jenkins Client 1.4	addon
mongodb-2.2	MongoDB NoSQL Database 2.2	addon
mysql-5.1	MySQL Database 5.1	addon

Figura 8.136: Cartridges disponibles

- 2.3. Ejecuta el siguiente comando para crear la aplicación **curso** con el *cartridge* **jbosses-2.0** (Tomcat7). Figura 8.137.

```
rhc app create curso jbosses-2.0
```

Observa la información que ofrece la salida del comando. Puedes observar que se ha creado la aplicación y se ha clonado el repositorio git (recuerda que la práctica anterior lo hicimos nosotros con el comando **git clone**).

- 2.4. Si accedes a tu cuenta con la consola de administración web puedes observar la información de tu nueva aplicación, Figuras 8.138 y 8.139.

3. Desplegar la aplicación

- 3.1. Accede a **Inicio**, **Todos los programas**, **Git**, **Git Bash** para abrir una consola de Git.
- 3.2. Accede al directorio de la aplicación (**cd curso**), Figuras 8.139 .

Observa que existen los directorios **src** y **webapps**, y el fichero **pom.xml**. El directorio **src** contiene la estructura en la que incluir el código fuente de la aplicación, y el fichero **pom.xml** es un fichero de configuración de **Maven** (<http://maven.apache.org/>) que describe la estructura de la aplicación y cómo construirla. El directorio **webapps** contendrá la aplicación desplegada.

En nuestro caso vamos a realizar el despliegue de una aplicación ya creada (fichero **war**) por lo que vamos a eliminar el directorio **src** y el fichero **pom.xml**. De esta forma le decimos a *OpenShift* que despliegue la aplicación a partir de nuestro fichero **war** y no a partir del código fuente.

En el siguiente enlace <https://github.com/openshift/origin-server/blob/master/cartridges/openshift-origin-cartridge-jbosses/README.md> puedes leer información sobre las diferentes forma de desplegar la aplicación.

- 3.3. Sobre el directorio raíz de la aplicación (**curso**) ejecuta el siguiente comando para borrar del repositorio Git el directorio **src** y el fichero **pom.xml**.

```
git rm -r src/ pom.xml
```

- 3.4. Copia el fichero **curso.war** en el directorio **webapps** con el nombre **ROOT.war**.


```

C:\Users\alumno>rhc app create curso jbosssews-2.0
DL is deprecated, please use Fiddle
Application Options
-----
Domain:      daw01
Cartridges:  jbosssews-2.0
Gear Size:   default
Scaling:     no

Creating application 'curso' ... done

Waiting for your DNS name to be available ... done

Cloning into 'curso'...
The authenticity of host 'curso-daw01.rhcloud.com (23.22.151.214)' can't be esta
blished.
RSA key fingerprint is cf:ee:77:cb:0e:fc:02:d7:72:7e:ae:80:c0:90:88:a7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'curso-daw01.rhcloud.com,23.22.151.214' (RSA) to the
list of known hosts.
remote: Counting objects: 41, done.
remote: Compressing objects: 100% (31/31), done.
rRemote: Total 41 (delta 2), reused 41 (delta 2)ceiving objects: 97% (40/41)
Receiving objects: 100% (41/41), 51.24 KiB | 0 bytes/s, done.

Resolving deltas: 100% (2/2), done.
Checking connectivity... done.

Your application 'curso' is now available.

URL:      http://curso-daw01.rhcloud.com/
SSH to:    53a54363e0b8cdfaf8000ba5@curso-daw01.rhcloud.com
Git remote: ssh://53a54363e0b8cdfaf8000ba5@curso-daw01.rhcloud.com/~/.git/curso
.git/
Cloned to: C:/Users/alumno/curso

Run 'rhc show-app curso' for more details about your app.
C:\Users\alumno>_

```

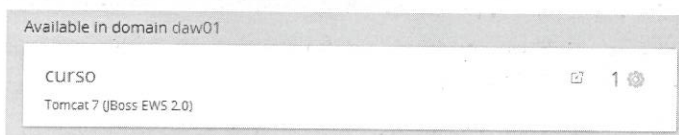
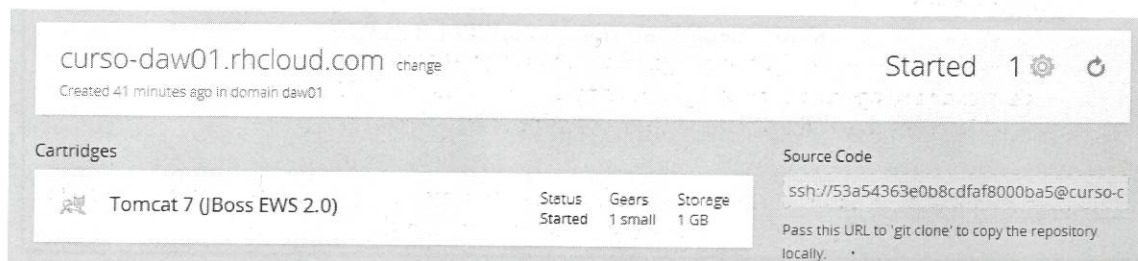
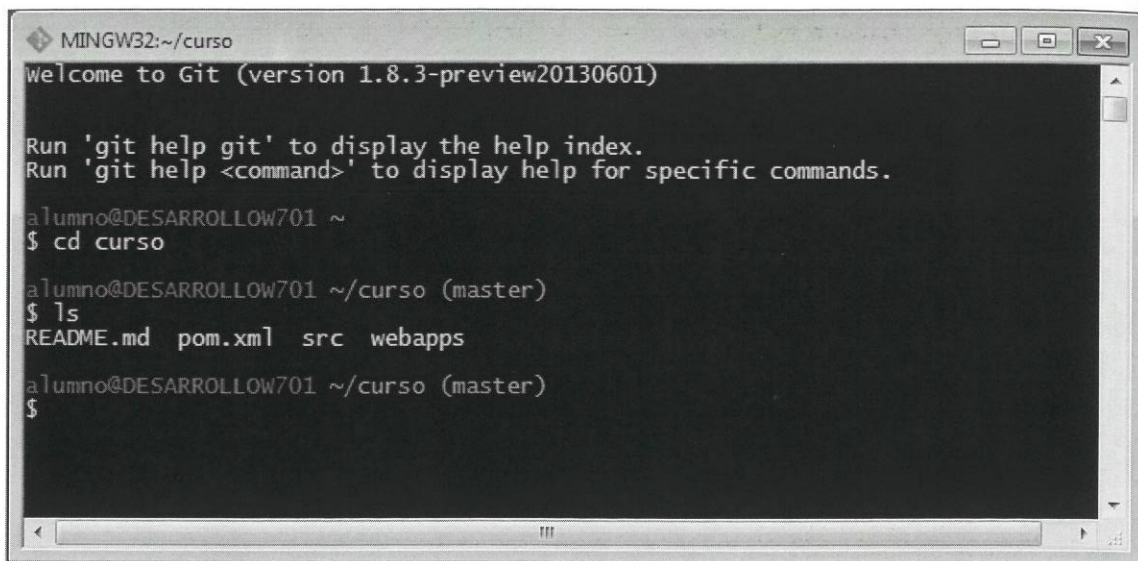
Figura 8.137: Creación de la aplicación **curso** con el *cartridge* jbosssews-2.0

Figura 8.138: Aplicaciones

Figura 8.139: Aplicación **curso**

A terminal window titled 'MINGW32:~/curso' showing the output of Git commands. The text inside the terminal is as follows:

```
MINGW32:~/curso
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

alumno@DESARROLLOW701 ~
$ cd curso

alumno@DESARROLLOW701 ~/curso (master)
$ ls
README.md  pom.xml  src  webapps

alumno@DESARROLLOW701 ~/curso (master)
$
```

Figura 8.140: Directorio de la aplicación

3.5. Sobre el directorio raíz de la aplicación (curso) ejecuta los siguientes comandos para subir los cambios al repositorio git de *OpenShift*, Figura 8.141.

```
git add -A
git commit -m "Primer despliegue de curso"
git push
```

3.6. Accede a la url de la aplicación **curso** para probar su funcionamiento, Figura 8.141.



Figura 8.141: Acceso a la aplicación **curso**

8.20. Despliegue en la nube (OpenShift) - 4

En esta práctica veremos cómo integrar *Eclipse* con *OpenShift* para poder desarrollar y desplegar aplicaciones.

1. Introducción al desarrollo con IDEs en *OpenShift*

En prácticas anteriores hemos utilizado *OpenShift Web Interface* y *OpenShift Command Line Interface* para desplegar aplicaciones en *OpenShift*. En esta práctica utilizaremos *Eclipse*.

Los siguientes enlaces contienen información útil de cómo integrar un IDE con *OpenShift*:

- <https://www.openshift.com/get-started/>
- <https://www.openshift.com/page/install-jboss-developer-studio>
- <https://www.openshift.com/blogs/getting-started-with-eclipse-paas-integration>

2. Instalación de *JBoss Tools*

- 2.1. Inicia sesión en **DesarrolloW7XX** con un usuario con privilegios de administrador.
- 2.2. Inicia *Eclipse* y accede a **Help, Eclipse Marketplace**. Busca *JBoss Tools*.
- 2.3. Instala la versión adecuada para tu *Eclipse*, Figura 8.142.

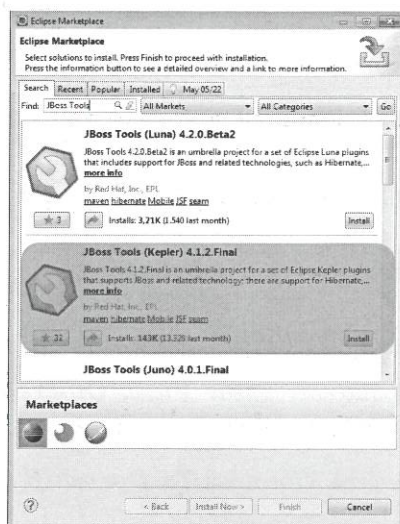


Figura 8.142: Instalación de *JBoss Tools*

2.4. Selecciona solo *JBoss OpenShift Tools* y pincha en *Next*, Figura 8.143.

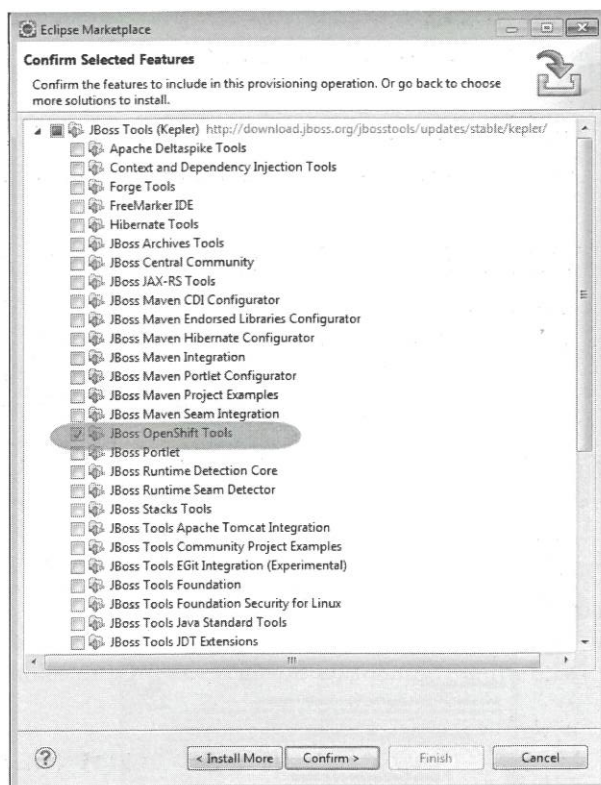


Figura 8.143: Instalación de *JBoss OpenShift Tools*

2.5. Cuando termine la instalación reinicia *Eclipse*.

3. Conexión con la cuenta de *OpenShift*

3.1. En Eclipse accede a *Windows, Show View, Others*.

3.2. Despliega *OpenShift* y selecciona *OpenShift Explorer*.

3.3. En la parte superior derecha *Connect To OpenShift* e introduce tus credenciales.

3.4. Accederás a tus aplicaciones, Figura 8.144.

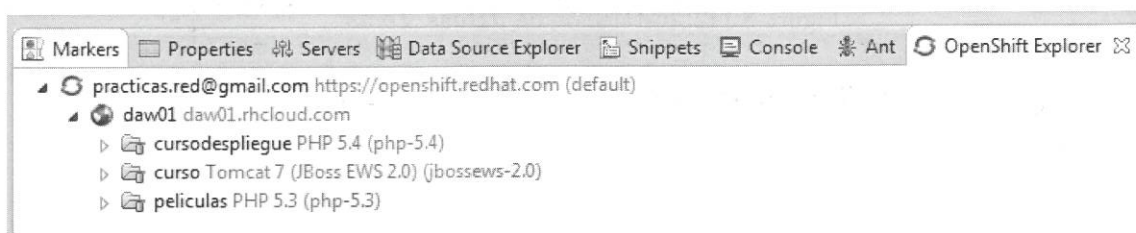


Figura 8.144: Aplicaciones en *OpenShift*

4. Crear una aplicación

- 4.1. Accede a *File, New, Others*.
- 4.2. Despliega *OpenShift* y selecciona *OpenShift Application*. Pincha en *Next* para continuar.
- 4.3. Introduce los datos de tu cuenta de *OpenShift* y pincha en *Next* para continuar.
- 4.4. Aquí podrías crear una nueva aplicación o importar alguna de tus aplicaciones, Figura 8.145.

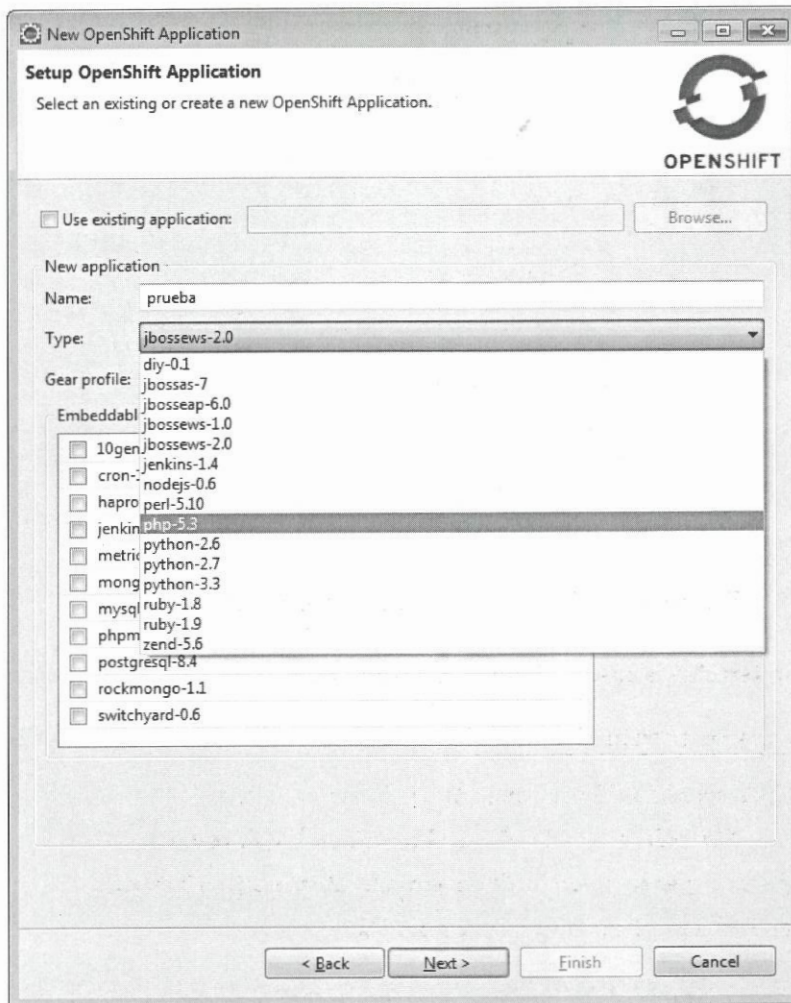


Figura 8.145: Crear una aplicación *OpenShift*