



UT05_00 JDBC Resources and Connection Pools

1. Crear recursos JDBC de MySQL en Glassfish

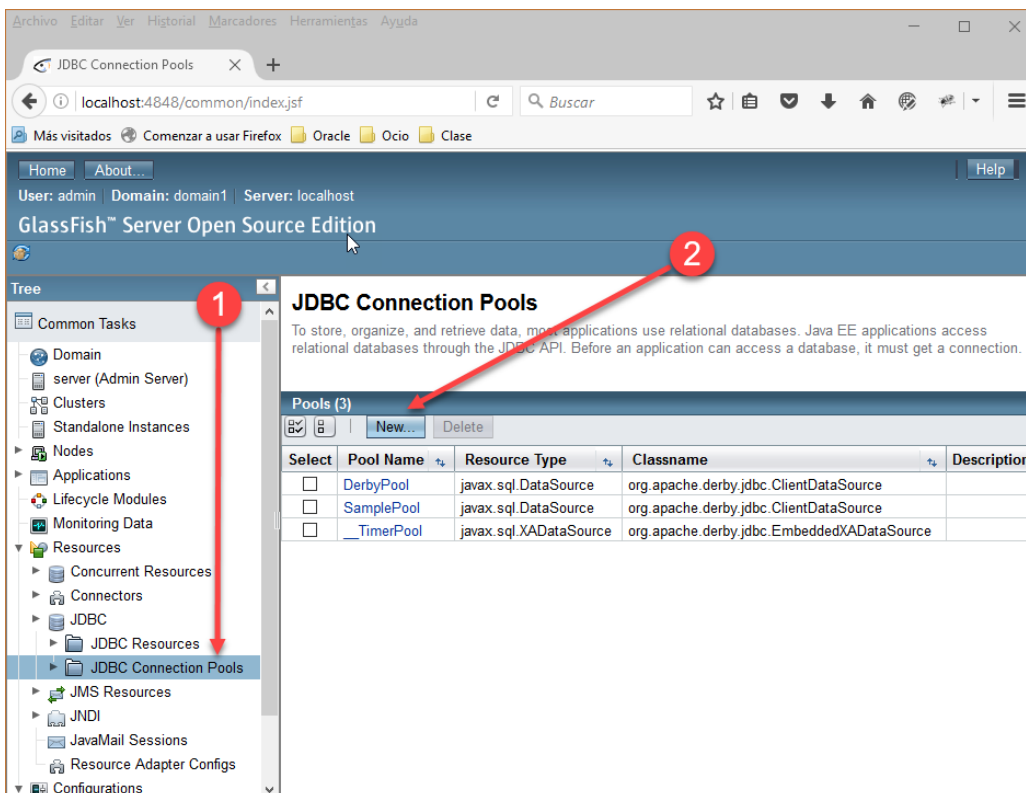
Vamos a crear un objeto DataSource que es una factoría de conexiones para un origen de datos que este objeto representaría. Constituye una alternativa al DriverManager y es la opción preferida de obtener conexiones, dado que éstas pasarán a estar gestionadas por el contenedor de servlets, introduciendo conceptos tales como pools de conexiones.

Un pool de conexiones es un conjunto de conexiones de base de datos, que el contenedor mantiene abiertas, y que son concedidas a las aplicaciones según las van demandando. El contenedor gestiona estas conexiones de forma más eficiente que si fueran las propias aplicaciones las encargadas de la gestión, ya que no cierra automáticamente la conexión cuando aplicación termina, sino que puede mantener la conexión activa, para darla a otra aplicación que más adelante pueda necesitar acceso JDBC.

1.1 Crear pool de conexiones

NOTA: Realmente no vamos a crear un pool de conexiones, pero sí un Datasource.

1. Ir a: **Consola Glassfish > Resources > JDBC > JDBC Connection Pools**
2. Pulsar: **New...**





3. Introducir las propiedades del Connection Pool

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

Next Cancel

* Indicates required field

General Settings

Pool Name: * MySQLConnPool

Resource Type: javax.sql.DataSource
Must be specified if the datasource class implements more than 1 of the interface.

Database Driver Vendor: MySql
Select or enter a database driver vendor

Introspect: ☐ Enabled
If enabled, data source or driver implementation class names will enable introspection.

1.1.1 BUG Glassfish 4.1.1

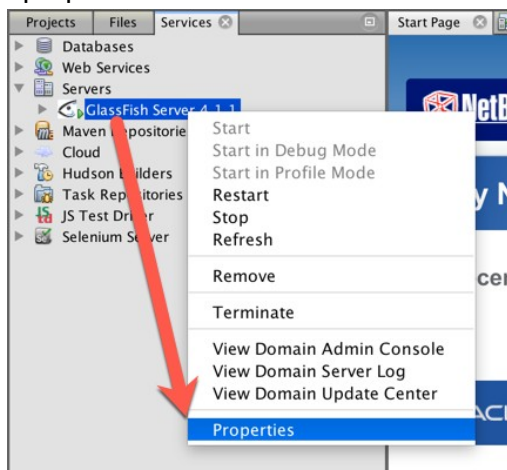
NOTA: Hay un BUG gordo en Glassfish 4.1.1 que implica la necesidad de crear a mano el pool de conexiones

<https://docs.oracle.com/cd/E19776-01/820-4494/ghsgw/index.html>

1.1.1.1 Localizar el comando asadmin

Vamos a tener que usar el comando **asadmin**. Vamos a ver cómo localizar la ruta donde está el comando.

1. Vamos a propiedades del servidor GlassFish



2. Buscamos la ruta del directorio de instalación (Installation Location)

1. La que nos interesa es la primera que aparece: **Installation Location**
(marcada con 1 en la imagen)
2. La segunda ruta **Domains Folder**, (marcada con 2 en la imagen) la utilizaremos más adelante, cuando copiamos el JAR de MySQL al CLASSPATH del servidor Glassfish.



Server Name: GlassFish Server 4.1.1
Server Type: GlassFish Server 4.1.1

Common Java

Installation Location: /Applications/NetBeans/glassfish-4.1.1/glassfish

Domains Folder: /Applications/NetBeans/glassfish-4.1.1/glassfish/domains

Host: localhost Loopback

DAS Port: 4848 HTTP Port: 8080

Domain: domain1 Target:

User Name: admin Password:

☐ Enable Comet Support ☐ Show Password in this Form
☐ Enable HTTP Monitor ☒ Preserve Sessions Across Redeployment
☒ Enable JDBC Driver Deployment ☒ Start Registered Derby Server

En mi caso, al ser OSX, la **Installation Location** es: **/Applications/NetBeans/glassfish-4.1.1/glassfish**

En Windows sería algo como **C:\Program Files\glassfish-4.1.1** o similar

Dentro de esta carpeta hay una carpeta **bin**, y dentro de ella está el **asadmin**. En mi caso el **asadmin** estaría en: **/Applications/NetBeans/glassfish-4.1.1/glassfish/bin/asadmin**

3. Puedo hacer:

```
cd /Applications/NetBeans/glassfish-4.1.1/glassfish/bin
```

y una vez aquí lanzar **asadmin** para crear el Pool.

1.1.1.2 Usar **asadmin** para crear el Pool

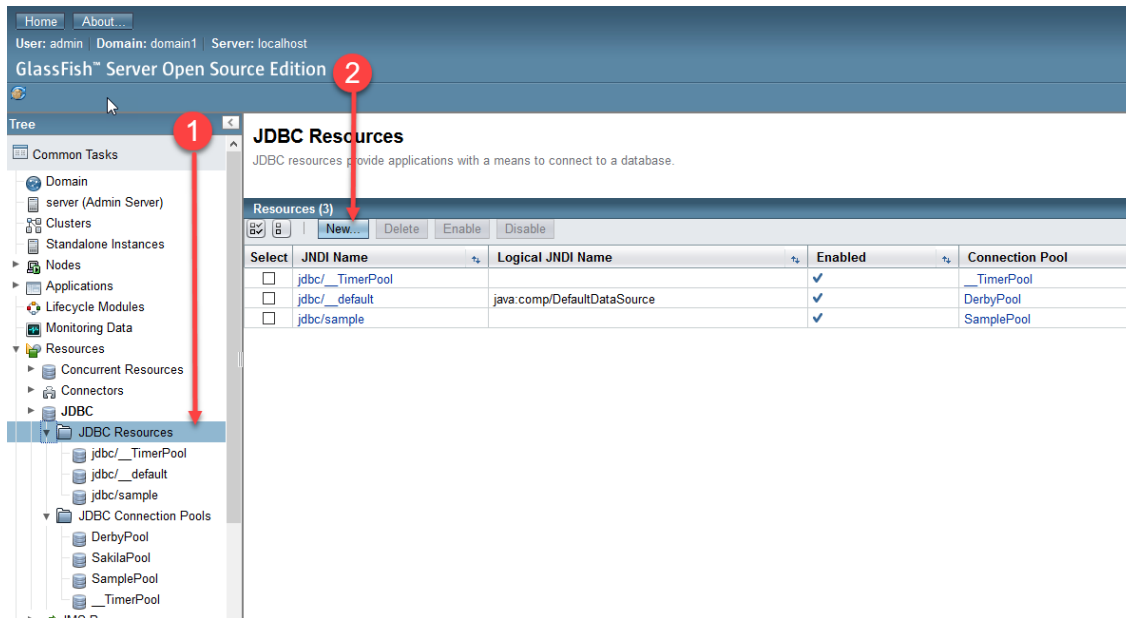
El comando para crear el pool desde línea de comandos:

```
asadmin create -jdbc-connectio-pool  
--datasourceclassname=com.mysql.jdbc.jdbc2.optional.MysqlDataSource  
--restype=javax.sql.DataSource  
--property=user=2daw:password=2daw:url="jdbc:mysql://localhost:3306/sakila" SakilaPool
```



1.2 Crear recurso JDBC

1. Ir a **Consola de administración > Resources > JDBC > JDBC Resources**
2. Pulsar **New...**



3. Introducir propiedades del recurso JDBC.

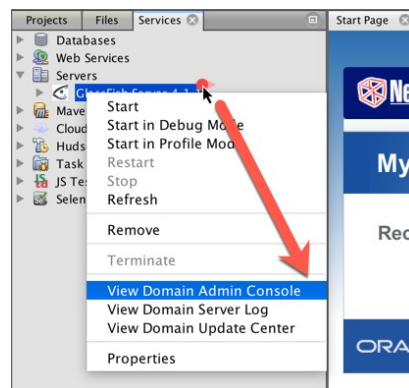
1.2.1 BUG Glassfish 4.1.1

NOTA: Hay un BUG gordo en Glassfish 4.1.1 que implica la necesidad de crear a mano el recurso JDBC.

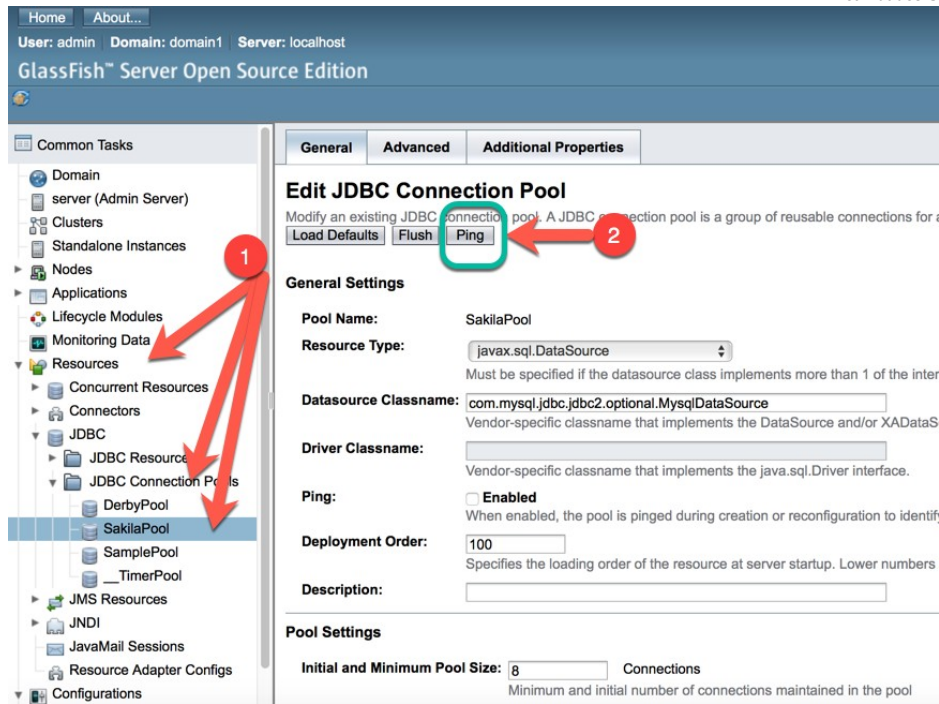
```
asadmin create-jdbc-resource --connectionpoolid SakilaPool jdbc/sakila
```

1.3 Verificar si funciona el Pool o falla (si falla seguramente falte el JAR de MySQL)

1. Abro la consola del servidor GlassFish: Right Click sobre el servidor y seleccionar **“View Domain Admin Console”**



2. Voy a **Resources > JDBC > JDBC ConnectionPools** y selecciono **SakilaPool**
3. En la parte derecha en la pestaña **General**, pulso el botón **Ping**, debería mostrarse un mensaje de OK, indicando que el Pool ha conectado con la base de datos



Mensaje de Ping OK, sería como el siguiente:



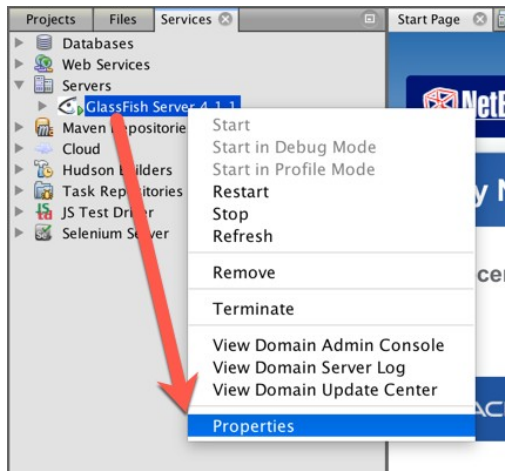
4. En caso de que no sea así, tenemos un problema... Seguramente sea porque Glassfish, no encuentra en el **CLASSPATH** !!!!! el JAR con la implementación del driver de MySQL (el JAR que utilizábamos nosotros cuando conectábamos con Driver Manager.

Vamos a ver cómo solucionar esto.

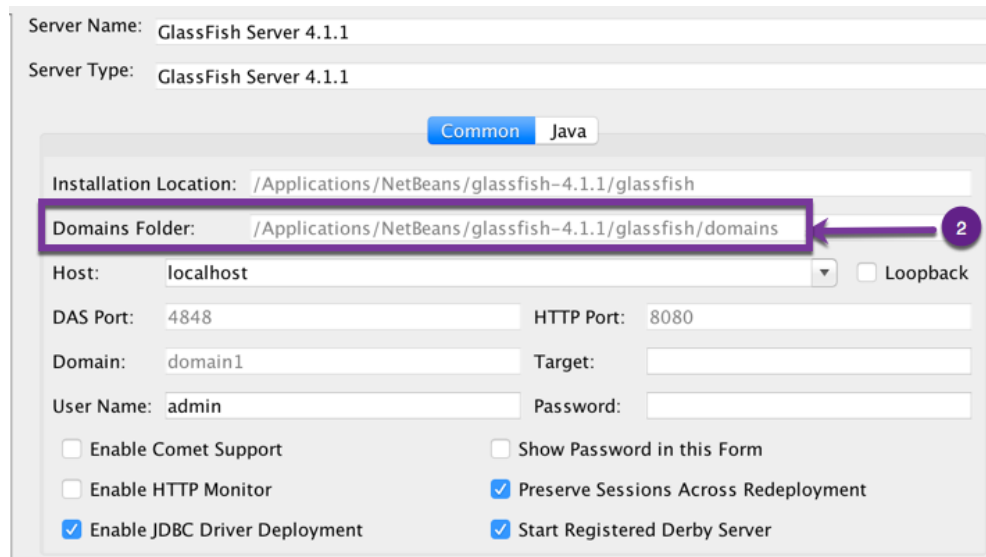
1.4 Solucionar problema de que GlassFish no encuentra el JAR con el driver de MySQL

Tenemos que localizar la ruta donde Glassfish va a ir a buscar el driver. Vamos a utilizar una ruta dentro de **Domains Folder**

1. Vamos a propiedades del servidor GlassFish



2. Buscamos la ruta del directorio de raíz de dominio (**Domains Folder**)



El que aparece marcado con (2)

3. Dentro de esa ruta hay una subcarpeta **domain1** y dentro de ella una **lib**. Es en esta carpeta lib donde hay que copiar el JAR de MySQL.

En mi caso sería en: /App...NetBeans/glassfish-4.1.1/glassfish/domains/**domain1/lib**

4. Ya, pero... ¿Dónde encuentro el JAR de MySQL?

Maven sabemos que utiliza un repositorio central de paquetes, pero es bien cierto que cada vez que descarga un paquete, lo almacena en un repositorio local.

Este repositorio local suele estar en el perfil de usuario, en una subcarpeta **.m2**

En Windows sería algo como C:\Users\profesor\.m2

Aquí verán una carpeta **repository** y dentro todos los JAR que nos hemos ido bajando.