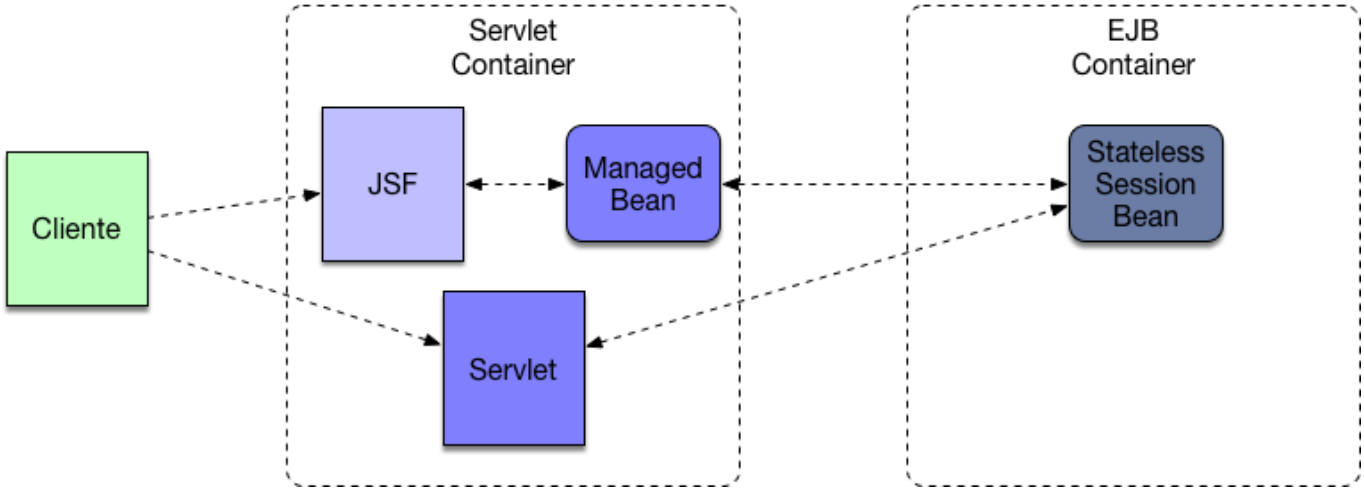
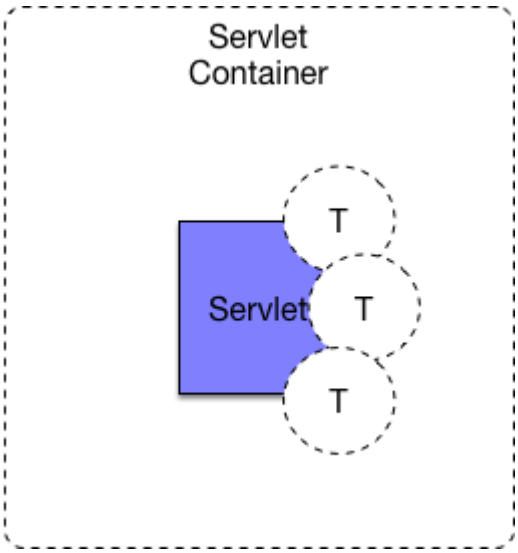
Los Enterprise Java Beans son a día de hoy componentes fundamentales en el desarrollo de aplicaciones Java Enterprise Edition . Sin embargo muchas veces surgen dudas sobre como funciona un Enterprise Java Bean a detalle. Todos tenemos bastante clara la idea básica y es que normalmente un cliente realiza una petición al servidor, el servidor la procesa a través de un Servlet o un Managed Bean (JSF) delegando en un EJB . El EJB recibe la petición invoca la funcionalidad de negocio correspondiente devolviendo un resultado que el Servlet/ Managed Bean termina presentando en una vista JSP /JSF page. Todo es muy sencillo o por lo menos lo parece.

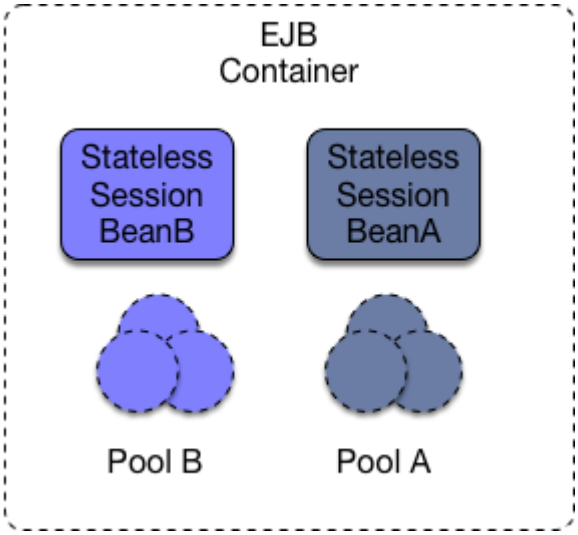


Enterprise Java Beans y Pools

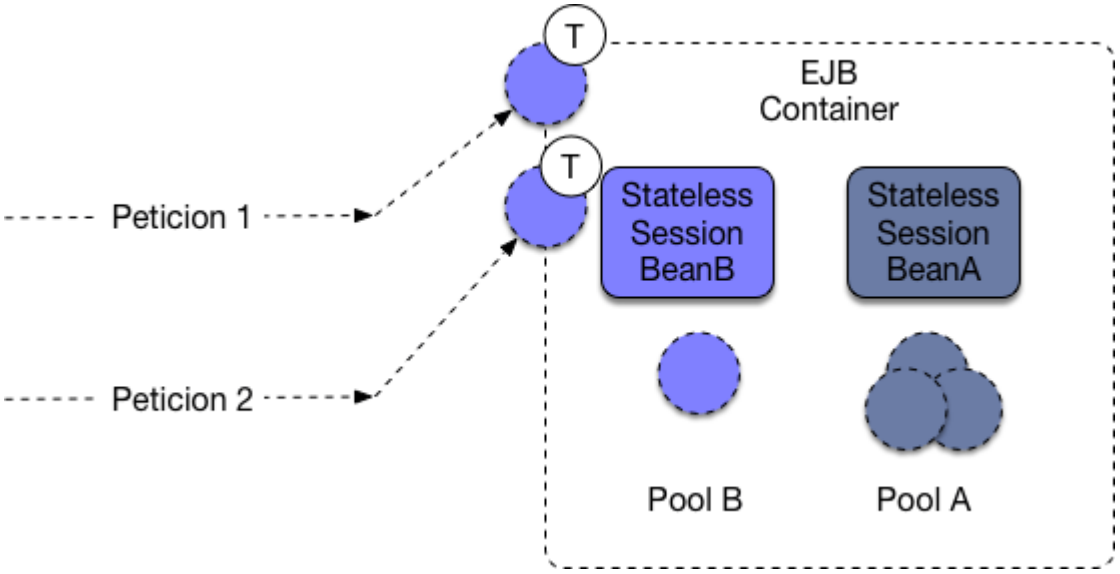
Los problemas comienzan con las preguntas más elementales. ¿Cuantas instancias de Servlet tenemos para gestionar varias peticiones de forma simultanea? . La realizad es que un servlet esta diseñado fundamentalmente para no almacenar estado y poder ser accedido de forma concurrente existiendo una única instancia del Servlet a nivel de Servlet Container.



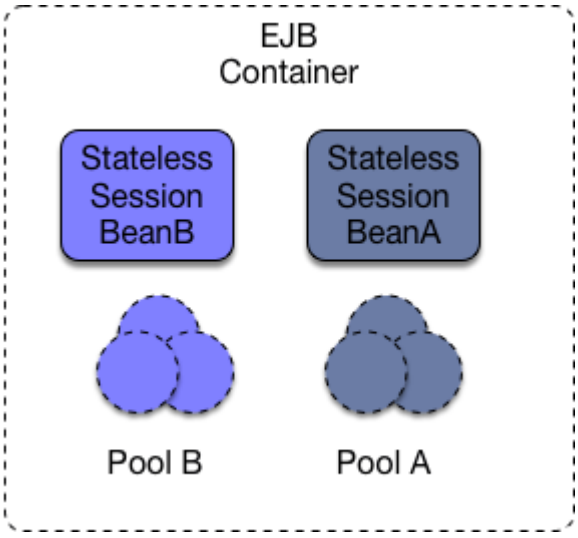
¿Sucede lo mismo a nivel de EJBs? . ¿Existe un único EJB que se encarga de procesar todas las peticiones concurrentes?. La realidad es que no . Los Enterprise Java Beans son configurados a través del concepto de Pool . Lo que implica que por cada EJB tipo de EJB existe un pool de EJBs que esta disponible.



Cuando al EJB container le llega una petición desde un Cliente (Servlet Container , o Swing) . Lo que hace el contenedor es sacar un EJB del pool y procesar la petición del cliente.

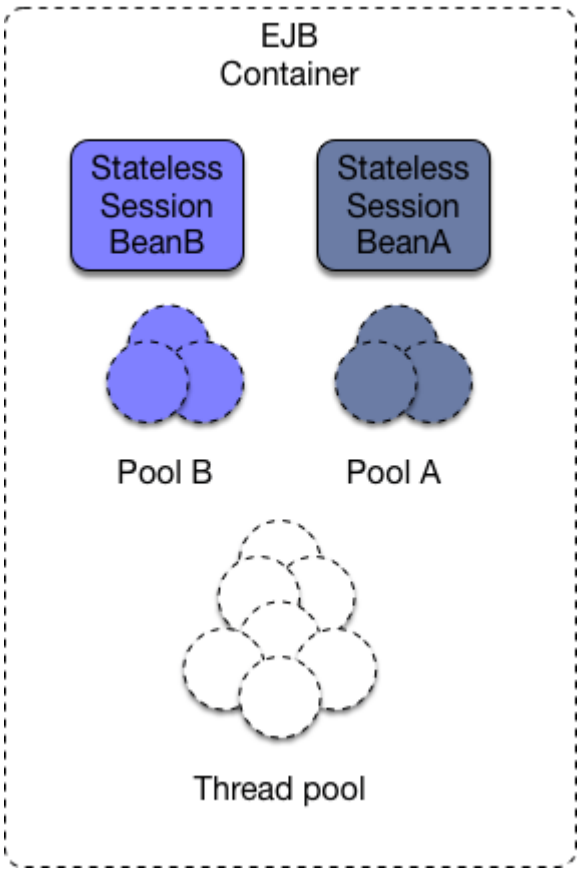


De esta forma se optimiza el número de EJBs disponibles y nos aseguramos que reducimos los problemas de concurrencia. Ya que cada EJB trabaja con un único thread luego estos regresan al pool.



Para procesar estas peticiones el EJB container tendrá que apoyarse en su pool de threads

ya que cada EJB obtendrá un hilo de ejecución distinta.



Es así como funciona un Enterprise Java Beans tengamoslo en cuanta cuando trabajemos con ellos. Esto es aplicable a Stateless Session Beans y Message Driven Beans que son los más utilizados . Los Statefull Session Beans y los EJB Singletons tienen comportamientos diferentes.

Otros artículos relacionados

1. El concepto de EJB in WAR y su uso
2. El concepto de EJB Event y como desacoplar servicios
3. Introducción a EJB 3.1 (I)
4. JSR