Servlet Container

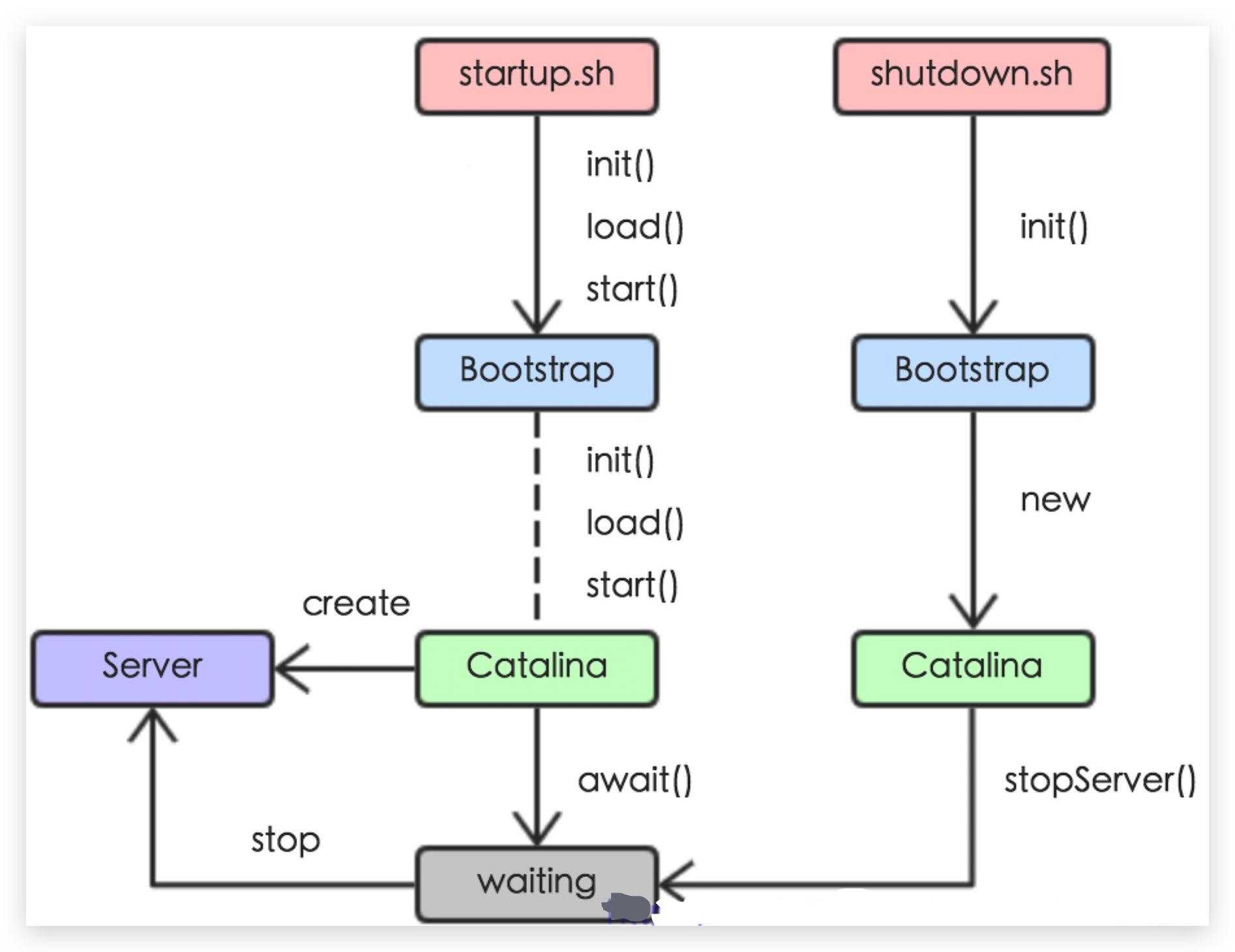
**基本思路**

Socket, NIO, ThreadPool, 协议解析, 路由分发, servlet

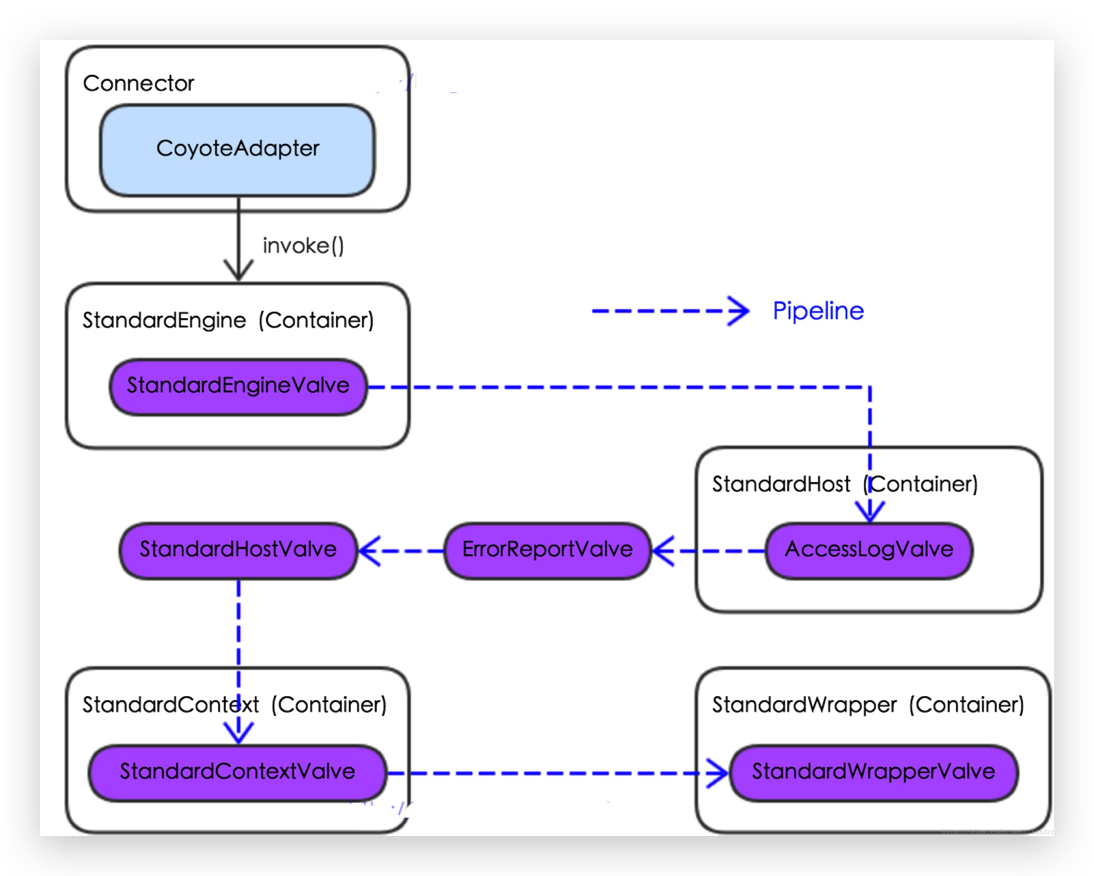
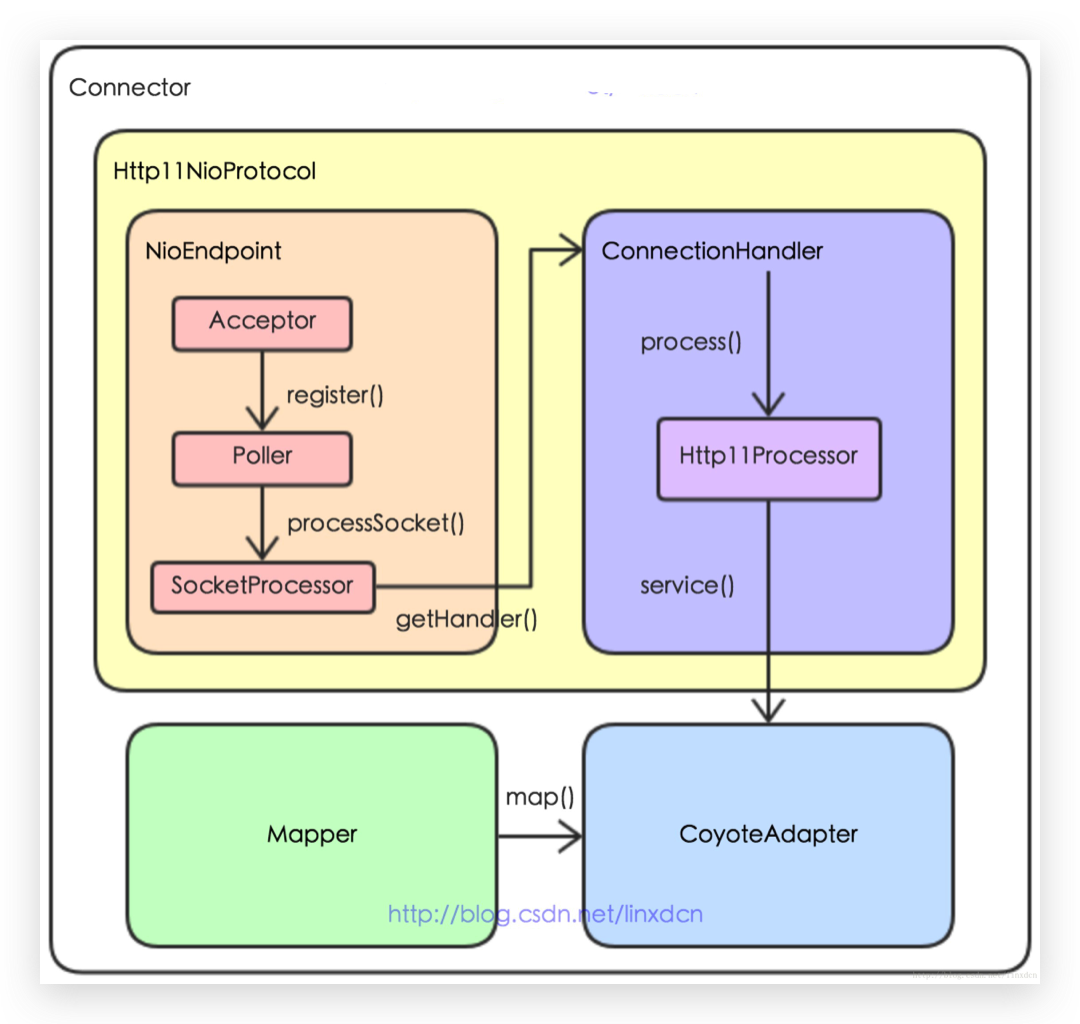
**组件(server.xml)**

|  |  |
| --- | --- |
| Lifecycle | 组件的生命周期：init、start、stop、destroy |
| Server | Service的容器 |
| Service | Engine和Connector的容器 |
| Connector | 解析某种网络协议的连接器，解析后转发给相应的Engine处理。 |
| Engine | 接收不同Connector的请求，处理后将结果返回给Connector。 |
| Host | 虚拟主机，/webapps |
| Context | 业务应用上下文，/webapps/ROOT |
| Wrapper | 封装servlet【约定：Servlet容器将Servlet类载入内存，并产生Servlet实例和调用service方法。**在一个进程中，每个Servlet类通常只有一个实例**。】（web.xml） |

**启停**

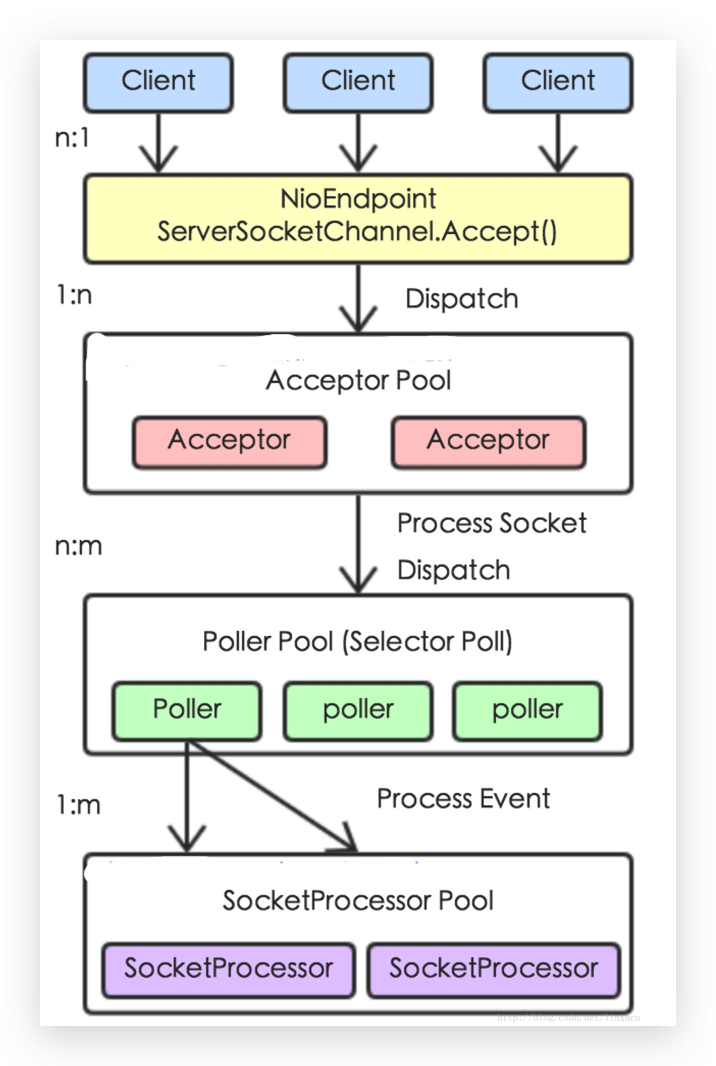


**请求-响应**

****

**线程模型**

服务端用于接收客户端连接的不再是1个单独的NIO线程，而是一个独立的NIO线程池

****

**HttpServlet.java**

****