TLS considerations

The HTTP/2 spec and the browser implementations bring new security constraints compared to existing secure HTTP/1.1 applications:

- TLS 1.2, SNI and ALPN, all required to upgrade to HTT/2 protocol
- a secure server certificate - usually with a strong signature algorithm and a key 2048+ bits key
- all the [TLS requirements listed in the HTTP/2 specification](#)

TLS 1.2 isn't natively supported by JDK8, but it is in JDK8u252+ and JDK9; also, alternative TLS implementations (including native bindings) are popular because they can offer performance gains compared to the JDK stack. This part explains why containers offer various deployment options and each has its pros/cons.

Container configuration

Apache Tomcat

As of version 8.5, Tomcat supports HTTP/2 with both JDK8 (using Tomcat Native) and JDK9 (with native JSSE). Servlet 4.0 is supported as of Tomcat 9.

If you'd like to use native bindings (Tomcat Native and OpenSSL), please follow [Tomcat's installation instructions](#) or use you favorite package manager to install those libraries. Note that you should make sure to use the right Tomcat Native and OpenSSL versions, compatible with the version of your Tomcat server.

Then you need to configure your [Tomcat Connector accordingly](#) (using the `Http11NioProtocol` and `JSSEImplementation` should work in most cases) and [configure HTTP/2 as an upgrade protocol](#).

Eclipse Jetty

Jetty supports several deployment modes, offering different ways to support TLS 1.2 and ALPN:

- by patching JDK8's ALPN implementation using a boot classpath jar; see [Jetty ALPN Agent](#) and [the ALPN reference doc chapter](#)
- by using native bindings with [Conscrypt](#) - see [the official blog post](#) for more on this

If you're running Jetty in standalone mode (i.e. Jetty's distribution), [the reference documentation is quite explicit](#).

If you're using Jetty as an embedded server, the [samples folder in Jetty's repository](#) has an HTTP2Server example.

Undertow

With Undertow 1.3, developers needed to use [Jetty's ALPN Agent](#) to run their server with ALPN support. As of Undertow 1.4, you can enable HTTP/2 support with a single option (see [reference documentation](#)).

Reactor Netty

As of Spring Framework 5.1 (Reactor Netty 0.8), this server supports as well HTTP/2. JDK9+ deployments will support that protocol without specific infrastructure changes.

For JDK 8 environments, or for optimal runtime performance, this server also supports HTTP/2 with native libraries. To enable that, your application needs to have an additional dependency.

Spring Boot manages the version for the `io.netty:netty-tcnative-boringssl-static` "uber jar", containing native libraries for all platforms. Developers can choose to import only the required dependendencies using a classifier (see the [Netty official documentation](#)).