Using CXF and CDI 1.1/1.2 (JSR-346) in OSGi environment

Introduction

Since 3.1.0 release, Apache CXF supports CDI-based deployment inside OSGi container which uses Pax CDI (Contexts and Dependency Injection for OSGi) implementation (https://github.com/ops4j/org.ops4j.pax.cdi). Apache CXF provides the **cxf-jaxrs-cdi** feature (for easy deployment into OSGi containers such as **Apache Karaf**) as well as **cxf-integration-cdi** module, which includes necessary OSGi bundle metadata to be used by Pax CDI.

Installation

For the installation steps, **Apache Karaf 3.0.3** is going to be used as it is the most widely used OSGi container. All commands shown below are intended to be executed from Apache Karaf shell. The first step is to install **http** and Pax CDI features, which is CDI-version dependent.

For CDI 1.1, please use **pax-cdi-1.1-web-weld** feature:

```
feature:install http pax-cdi-1.1-web-weld
```

For CDI 1.2, please use **pax-cdi-1.2-web-weld** feature:

```
feature:install http pax-cdi-1.2-web-weld
```

Next, Apache CXF 3.1.0+ features should be installed:

```
feature:repo-add cxf 3.1.0
feature:install cxf/3.1.0 cxf-jaxrs-cdi/3.1.0
```

Bundle Metadata

In order for the bundle to be recognized as web CDI one and use Apache CXF CDI capabilities, it should provide special bundle manifest instructions (f.e. by using **maven-bundle-plugin** plugin).

```
</instructions>
    ...
    <Import-Package>
        javax.servlet;version="[2.6,4)",
        org.apache.cxf.jaxrs;version="[3.1,4)",
        org.apache.cxf.cdi;version="[3.1,4)",
        *
    </Import-Package>

    <Require-Capability>
        org.ops4j.pax.cdi.extension; filter:="(&amp;(extension=cxf-integration-cdi))",
        osgi.extender; filter:="(osgi.extender=pax.cdi)"
    </Require-Capability>
    <Web-ContextPath>...</Web-ContextPath>
    <_wab>src/main/webapp</_wab>
</instructions>
```

The **Require-Capability** instruction is very important in order for CDI initialization, discovery and injections to work with Apache CXF. The **Web-ContextPath** is the context path for this web application to be deployed at. And **_wab** is the instruction to point out the **web.xml** file location. Please notice, the application will not be deployed under usual **/cxf** endpoint (common endpoint for regular Apache CXF services). The reason for that is that Pax Web is used for deployment of web CDI applications.

Web Application Configuration

At the moment, Apache CXF OSGi application should explicitly provide **web.xml** descriptor with at least **CXFCdiServlet** defined. For example:

```
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <servlet>
        <servlet-name>CXFServlet</servlet-name>
        <display-name>CXF Servlet</display-name>
        <servlet-class>org.apache.cxf.cdi.CXFCdiServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>CXFServlet</servlet-name>
        <url-pattern>/*</url-pattern>
    </servlet-mapping>
</web-app>
```

Discovery

Pax CDI uses own definition of bean bundles and as such, only bundles which follow these requirements are available for discovery (providers, services and features) and injection (see please https://ops4j1.jira.com/wiki/display/PAXCDI/Getting+Started+for+OSGi+Users for more details).