Scenario 1: Install Liberty on GCP Linux VM and Deploy Hello World WAR

Step-by-Step Setup

Install Liberty on GCM VM

```
→ wget
https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/wasdev/downloads/w
lp/latest/wlp-base-java8-linux.zip
→ sudo apt install unzip default-jdk -y
→ unzip wlp-base-java8-linux.zip -d /opt/
→ export WLP_HOME=/opt/wlp
→ export PATH=$WLP_HOME/bin:$PATH
```

Create Liberty Server

```
→ server create helloCubenSquare
→ cd $WLP_HOME/usr/servers/helloCubenSquare
```

Update `server.xml`

```
<server description="Hello Liberty">
  <featureManager>
    <feature>servlet-4.0</feature>
  </featureManager>
  <httpEndpoint host="*" httpPort="9080" />
  <webApplication location="hello.war" contextRoot="/hello" />
</server>
```

Create Hello World Java App

```
→ vi HelloServlet.java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
   IOException {
    res.setContentType("text/html");
    res.getWriter().println("<h1>Hello, Liberty!</h1>");
  }
}
```

Build WAR File :

Folder Structure -

```
hello/
├── WEB-INF/
│   ├── classes/
│   │   └── HelloServlet.java
│   └── web.xml
```

```
→ mkdir -p hello/WEB-INF/classes
→ cp HelloServlet.java hello/WEB-INF/classes/
→ cd hello/WEB-INF/classes
→ javac HelloServlet.java
→ cd ../..
→ echo '<web-app><servlet><servlet-name>HelloServlet</servlet-name><servlet-
   class>HelloServlet</servlet-class></servlet><servlet-mapping><servlet-
   name>HelloServlet</servlet-name><url-pattern>/</url-pattern></servlet-
   mapping></web-app>' > WEB-INF/web.xml
→ jar cvf hello.war *
→ cp hello.war $WLP_HOME/usr/servers/helloServer/dropins/
```

Start Liberty

```
server start helloServer
```

Access in Browser

```
http://<GCP_VM_EXTERNAL_IP>:9080/hello
```

Scenario 2: Create Liberty Base Image with Custom `server.xml`

Files Required

```
→ vi server.xml
<server description="Liberty Base">
  <featureManager>
    <feature>servlet-4.0</feature>
  </featureManager>
  <httpEndpoint host="*" httpPort="9080" />
</server>

→ vi Dockerfile:

FROM icr.io/appcafe/websphere-liberty:kernel-java17-openj9-ubi
```

```
COPY --chown=1001:0 server.xml /config/
RUN configure.sh
```

Build Base Image

```
docker build -t liberty-base .
```

Scenario 3: Deploy WAR in Liberty Base Image and Run Container

Files Required

- Dockerfile:
- FROM liberty-base
- COPY --chown=1001:0 hello.war /config/dropins/
- RUN configure.sh
- hello.war:
- Use the WAR built in Scenario 1.

Build and Run

```
docker build -t liberty-hello .
docker run -d -p 8080:9080 liberty-hello
```

Access in Browser

```
http://localhost:8080/hello
```

Scenario 4 : Update the Java Servlet (Change Background Color)

Modify your **`HelloServlet.java`** like this:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
IOException {
    res.setContentType("text/html");
    res.getWriter().println("<html><body style='background-color:#f0f8ff;'>");
    res.getWriter().println("<h1>Hello, Liberty!</h1>");
    res.getWriter().println("</body></html>");
  }
}
```

This sets the background color to a soft blue (`#f0f8ff`). You can change it to any hex code or named color.

Step 2: Rebuild the WAR File

Assuming your folder structure is still:

```
hello/
├── WEB-INF/
│   ├── classes/
│   │   └── HelloServlet.java
│   └── web.xml
```

Run:

```
→ cd hello/WEB-INF/classes
→ javac HelloServlet.java
→ cd ../..
→ jar cvf hello.war *
```

Step 3: Update the Docker Image

Update your Dockerfile if needed:

```
FROM liberty-base
COPY --chown=1001:0 hello.war /config/dropins/
RUN configure.sh
```

Then rebuild and redeploy:

```
docker build -t liberty-hello:v2 .
docker stop <old_container_id>
docker rm <old_container_id>
docker run -d -p 8080:9080 liberty-hello:v2
```

Step 4: Access the Updated App

Visit:

```
http://localhost:8080/hello
```

You should now see the same greeting with a new background color.
