Hackathon

2015-04-08 Hewlett-Packard

Contents

HP Helion 1.1 Development Platform: Java Hello World Sample

This sample demonstrates the minimum requirements for a functional Java application running on Helion OpenStack. This application displays the text "Hello World!" in a web page. You can use this sample to ensure that you have set up your environment for deployment to Helion Development Platform

Java Prerequisites

This section describes Java prerequisites for running the sample application.

Java Prerequisites - Windows

This topic describes the Java prerequisites for Microsoft Windows.

Install the appropriate Java for your platform:

1. Navigate to the *JDK installation page* and run the appropriate JDK installer for your chosen platform.
2. Navigate to the *Maven installation page* and follow the *installation instructions*.

Java Prerequisites - Mac and Unix

This topic describes the Java prerequisites for Macintosh OS X and Unix/Linux.

Install the appropriate Java for your platform:

From a command prompt, enter the following command:

```
sudo apt-get install default-jdk maven
```

Download the Application Files

Download and unzip the application:

```
sample command goes here
```

Note where you downloaded the file.

Build the Application

In the root directory of the sample package, enter the following command:

```
mvn clean package
```

This builds the application with Maven. It will create the `hello-world-java-1.0.war` file within the target directory.

Deploy the Application

Use the Helion client to deploy your app to Helion Development Platform.

Note: Application Lifecycle Service clusters that require an upstream HTTP proxy to access the internet will need to be *made aware of the proxy*. The sample applications require access to the Internet in order to download dependent packages.

1. Open the *Helion command-line interface (CLI).*
2. Target your desired environment. For example:
   ```
   helion target https://api.xx.xx.xx.xx.xip.io
   ```
3. Log in to your desired environment:
   ```
   helion login
   ```
4. Change to the root directory of the sample:
   ```
   cd helion-hello-world-java-master
   ```
5. Deploy the application:
   ```
   helion push
   ```
   Accept any default values that you may be prompted for.

   

   Note: By default, ALS clusters are configured with two domains (private and public). In some situations the Helion CLI may prompt you to select a target domain. If prompted, select the public domain from the given list (i.e. <app-name>.xxx.xxx.xxx.xxx.xip.io)

Key Code Snippets

This simple Servlet prints "Hello World".

```
package org.hp.samples;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

   protected void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
        response.setContentType("text/plain");
        response.setStatus(200);
        PrintWriter writer = response.getWriter();
        writer.println("Hello World");
        writer.close();
    }
}
```

The POM.xml file contains the configuration information generated by Maven and used by ALS to set up the environment.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.hp.samples</groupId>
    <artifactId>hello-world-java</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>
    <dependencies>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>servlet-api</artifactId>
            <version>2.5</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
</project>
```

Run the Application

1. Open the Helion Management Console. This is the web-based administrative interface that can be reached by typing the ALS endpoint URL into a browser window.
2. Click **Applications**.
3. If the file push was successful, you should see `hello-world-java` in the list of available applications.
4. The status of the application should be `Started`. Click the name of the application to launch it.
5. In the upper right-hand corner, click **View App**.
6. You should see a simple text message: `Hello World!`.

Key Learnings

ALS requires configuration information to create an environment for your app. Configuration information is contained in the pom.xml file. Tools like Maven can generate this configuration file for you.

Exit Samples

Previous Sample

Next Samples