IBM WebSphere Application Server V8.5 lab
Liberty profile administration using the command line

Scenario

You are a system administrator responsible for managing web application server installations. Your supervisor has asked you to deploy two sample applications to the WebSphere Application Server V8.5 Liberty profile so that you can help your team to more fully understand how to administer the offering.

Goals

The primary goal of this lab is to learn how to deploy and administer the Liberty profile using the command line.

You will do the following:

1. Deploy Liberty profile from an archive (self-extracting JAR file).
2. Deploy and test a simple servlet application.
3. Deploy and test a servlet application that requires basic web security configuration.
4. Configure a server to use non-default ports for HTTP and HTTPS traffic.
5. Move a portion of a server configuration to an included configuration file.
6. Explore additional server utility command line options.
7. Configure logging settings and dump problem determination to file.

For more information, see the following information center topics:

- The Liberty profile
- Installing the Liberty profile
- Setting up the Liberty profile application-server environment
- Administering the Liberty profile
- Securing the Liberty profile and its applications
- Deploying applications to the Liberty profile
- Monitoring the Liberty profile
- Liberty profile: Troubleshooting tips

This lab is provided **AS-IS**, with no formal IBM support.

Prerequisites

This lab requires a single host machine with the following software and materials:

- A copy of the WebSphere Application Server V8.5 Liberty profile archive (self-extracting JAR file). To download the archive, see the WASdev community downloads page. You can also use IBM Installation Manager and a WebSphere Application Server product repository to install the Liberty profile. For more information about both installation methods, see the WebSphere Application Server V8.5 information center topic Installing the Liberty profile.
- Java 6 installed as the default (system) Java Runtime Environment (JRE). The minimum supported level for the JRE from Oracle is Java™ 6 update 26. For the Java JRE from IBM, the minimum supported level is 6.0 (J9 2.6) SR 1. Java 7 is supported; however, there are several significant restrictions. For more information, see the WebSphere Application Server V8.5 information center topic Liberty profile: Runtime environment known restrictions.

- The lab materials file, `WASv85Labs_LibCmdLine.zip`. To download this file, visit the WebSphere Application Server V8.5 area of the IBM Education Assistant site.

Procedure

A. Getting started

Tasks

1. Extract lab materials file `WASv85Labs_LibCmdLine.zip` into a suitable directory. You can extract this file and any other WebSphere Application Server V8.5 lab materials files into the root directory. For example, on Windows, extracting the file into the `C:\` directory will create lab materials directory `C:\WASv85Labs\LibCmdLine`, and so on. You can also use the same basic approach on UNIX/Linux.

2. Install the Liberty profile to lab directory `WASv85Labs/LibCmdLine`

   The following instructions assume that you will be using a Liberty profile archive (self-extracting JAR file) to install the Liberty profile. You can also use IBM Installation Manager and a WebSphere Application Server product repository. For more information about both installation methods, see the WebSphere Application Server information center topic Installing the Liberty profile.

   a. Open a command prompt and change to the directory that contains the Liberty profile archive (self-extracting JAR file). Then run the following command:

   ```
   java -jar wlp-edition-8.5.0.0.jar
   ```

   b. Press **Enter** to view the license terms.

   c. Press **Enter** to view additional license information.

   d. Enter **1** to indicate that you agree to the terms of the license agreements. Then press **Enter**.

   e. Enter the full path to lab directory `WASv85Labs/LibCmdLine`, for example,

   Windows: `C:\WASv85Labs\LibCmdLine`
   UNIX/Linux: `/WASv85Labs/LibCmdLine`

   Then press **Enter** to extract the files to the target directory.

   After the installation completes, verify that lab directory `WASv85Labs/LibCmdLine/wlp` contains the standard Liberty profile directories `bin`, `clients`, `dev`, `lafiles`, `lib`, `templates`, and `usr`.

B. Deploy and test application ServletSample

Notes

- The application (`ServletSample.war`) is located in lab directory `WASv85Labs/LibCmdLine/apps`
- The application uses the Servlet programming model.

Tasks

1. Start the default server.

   Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   Then run the following command: `server start`

2. Deploy the ServletSample application to the default server.

   Copy `ServletSample.war` from lab directory `WASv85Labs/LibCmdLine/apps` to lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/defaultServer/dropins`

   For more information about deploying applications from the dropins directory, see the README.TXT file located in lab directory `WASv85Labs/LibCmdLine/wlp`, and the WebSphere Application Server Information center topic Deploying applications to the Liberty profile.

3. Examine the server logs to verify that the application started properly.

   Use a text editor to view log files `console.log` and `messages.log` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/defaultServer/logs`. You should see a message in each file indicating that application ServletSample started and a message listing the URL for accessing the application.

   Note that each time that the server is started, `console.log` is overwritten, whereas a new version of `messages.log` is created. Also note that the entries in `messages.log` include additional fields such as timestamps and component names.

4. Verify that the application properly returns a page that displays the date and time.

   Use a web browser to visit the following URL: `http://localhost:9080/ServletSample`

5. Stop server defaultServer.

   Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   Then run the following command: `server stop`

   Note that when a server name is not specified after an action (such as stop), the command will act upon the server defaultServer.

C. Deploy and test application SecuritySample

Notes

- The application (`SecuritySample.war`) is located in lab directory `WASv85Labs/LibCmdLine/apps`
- The application uses the JSP programming model.
- The application deployment descriptor is set up so that the following URL patterns are protected in the following manner:
  - /*
    - Users must authenticate and possess a role of administrator or user.
  - /admin
    - Users must authenticate and possess a role of administrator.
  - /user
    - Users must authenticate and possess a role of user.
  - /ssl
    - Users must authenticate and possess a role of administrator or user.
    - Use of SSL is mandatory.

- The application development team recommends configuring the server in the following manner so that it is possible to test the security features of the application:
  - Include a simple user registry that contains the following users with the following passwords and group assignments:

User	Password	Group
gkelly	gkelly1	teacher
svaughn	svaughn1	teacher
rkumar	rkumar1	student
mlee	mlee1	student

  - Define the application in a manner that maps the role administrator to the group teacher, and the role user to the group student.

Tasks

1. Create a new server named SecuritySampleServer.

   Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   Then run the following command: `server create SecuritySampleServer`

2. Configure SecuritySampleServer to support basic web application security and SSL connections.
   a. Using a text editor, open the server configuration file (`server.xml`) located in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`
   b. Extend `server.xml` to include the application security and SSL features. Then add a key store to support the SSL feature. For example:

      ```
      <featureManager>
        <feature>jsp-2.2</feature>
        <feature>appSecurity-1.0</feature>
        <feature>ssl-1.0</feature>
      </featureManager>

      <keyStore id="defaultKeyStore" password="dksPassword"/>
      ```

      Be sure to place all elements between the `<server>` and `</server>` tags.
   c. Encode the key store password. (Optional)
      i. Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`
      ii. Run the `securityUtility` command with the `encode` option to generate and display an encoded version of the key store password, for example, `securityUtility encode dksPassword`
      iii. Within the `<keystore>` element of the server configuration, replace the plain text password with the encoded password, for example,

         ```
         <keyStore id="defaultKeyStore" password="{xor}OzQsDz4sLCgwLTs="/>
         ```

3. Configure SecuritySampleServer to include the required simple user registry. (See the Notes section for details.)
   a. Extend `server.xml` to include a basic user registry containing the required users and groups, for example:

      ```
      <basicRegistry id="basic">
        <user name="gkelly" password="gkelly1"/>
        <user name="svaughn" password="svaughn1"/>
        <user name="rkumar" password="rkumar1"/>
        <user name="mlee" password="mlee1"/>
        <group name="teacher">
          <member name="gkelly"/>
          <member name="svaughn"/>
        </group>
        <group name="student">
          <member name="rkumar"/>
          <member name="mlee"/>
        </group>
      </basicRegistry>
      ```

      Be sure to place all elements between the `<server>` and `</server>` tags.
   b. Use the procedures outlined in step 2c to encode the password for each user. (Optional)

4. Configure SecuritySampleServer to include application SecuritySample. Within the application definition, map the role administrator to the group teacher, and role user to the group student.
   a. Copy `SecuritySample.war` from lab directory `WASv85Labs/LibCmdLine/apps` to lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/apps`
      Note: If you are using V8.5.0.0 of the Liberty profile server, you must create the `apps` directory for the server.
   b. Extend `server.xml` to include the SecuritySample application. Within the application definition, map the role administrator to the group teacher, and role user to the group students. For example:

      ```
      <application id="SecuritySample" name="SecuritySample"
      location="SecuritySample.war" type="war">
        <application-bnd>
          <security-role name="admin">
            <group name="teacher"/>
          </security-role>
          <security-role name="user">
            <group name="student"/>
          </security-role>
        </application-bnd>
      </application>
      ```

      Be sure to place all elements between the `<server>` and `</server>` tags.

5. Start SecuritySampleServer.

   Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   Then run the following command: `server start SecuritySampleServer`

6. Examine the server logs to verify that the application started properly.

   Use a text editor to view the `console.log` file in lab directory
   `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/logs`

   You should see a message indicating that application SecuritySample started, and a message listing the URL for accessing the application.

7. Verify the following:
   a. URL pattern /* can only be accessed by a user possessing the user or administrator role.
      i. Open a new web browser instance and visit
         `http://localhost:9080/SecuritySample`
      ii. To verify that the URL pattern is protected, attempt to log in with a user name or password that does not exist in the user registry.
      iii. To verify that the URL pattern can be accessed by an authorized user, log in as any one of the four users included in the user registry.
      iv. Close the web browser.
   b. URL pattern /admin can only be accessed by a user possessing the administrator role.
      i. To verify that the URL pattern is protected, open a new web browser instance and visit
         `http://localhost:9080/SecuritySample/admin`
         Then attempt to log in as a user possessing the user role (rkumar or mlee).
      ii. Close the web browser.
      iii. To verify that the URL pattern can be accessed by an authorized user, open a new web browser instance and visit `http://localhost:9080/SecuritySample/admin`
         Then log in as a user possessing the administrator role (gkelly or svaughn).
      iv. Close the web browser.
   c. URL pattern /ssl requires the use of SSL.
      i. To verify that it is possible to connect using HTTPS directly, open a new web browser instance and visit `https://localhost:9443/SecuritySample/ssl`
         When prompted, confirm the security exception. Then log in as any one of the four users included in the user registry.
      ii. To verify that HTTP requests are properly redirected to HTTPS and the HTTPS port, visit
         `http://localhost:9080/SecuritySample/ssl`
      iii. Close the web browser.

8. Add two new users to the user registry, one assigned to the group teacher, and another assigned to the group student. Then verify that one of the new users can access the SecuritySample application.
   a. Extend the user registry to include the new users and the new group assignments, for example:

```
<basicRegistry id="basic">
  <user name="gkelly" password="gkelly1"/>
  <user name="svaughn" password="svaughn1"/>
  <user name="lhess" password="lhess1"/>
  <user name="rkumar" password="rkumar1"/>
  <user name="mlee" password="mlee1"/>
  <user name="bchi" password="bchi1"/>
  <group name="teacher">
    <member name="gkelly"/>
    <member name="svaughn"/>
    <member name="lhess"/>
  </group>
  <group name="student">
    <member name="rkumar"/>
    <member name="mlee"/>
    <member name="bchi"/>
  </group>
</basicRegistry>
```

   b. Use the procedures outlined in step 2c to encode the password for each user. (Optional)
   c. Verify that one of the new users can access URL pattern /*.
      i. Open a new web browser instance and visit `http://localhost:9080/SecuritySample`
      ii. Log in as one of the new users.
      iii. Close the web browser.

D. Configure SecuritySampleServer to use new (non-default) ports for HTTP and HTTPS traffic

Tasks

1. Configure SecuritySampleServer to use new (non-default) TCP/IP ports for HTTP and HTTPS traffic. Store the port number values in custom properties (variables) in a server boot properties file, effectively separating the machine-specific settings from the server configuration.
   a. Stop SecuritySampleServer.

      Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

      Then run the following command: `server stop SecuritySampleServer`
   b. Create text file `bootstrap.properties` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`

      Then edit the file and add the following server boot properties:

```
default.http.port=9081
default.https.port=9444
```

c. Using a text editor, open the server configuration file (`server.xml`) located in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`

   Then change the HTTP endpoint definition to the following:

   ```
   <httpEndpoint id="defaultHttpEndpoint" host="localhost"
   httpPort="${default.http.port}" httpsPort="${default.https.port}"/>
   ```

   With these changes in place, SecuritySampleServer will use the HTTP and HTTPS port values that are stored in `bootstrap.properties`.

d. Start SecuritySampleServer.

   Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   Then run the following command: `server start SecuritySampleServer`

2. Review log messages related to the server configuration change.

   Use a text editor to view log file `console.log` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/logs`

   You should see a message that includes the updated URL for accessing the application.

3. Verify that the server is now using the new (non-default) HTTP and HTTPS ports.

   a. Open a new web browser instance and visit `http://localhost:9081/SecuritySample`

   b. Log in as any one of the six users.

   c. Visit `https://localhost:9444/SecuritySample/ssl` (Accept the security exception.)

   d. Close the web browser.

E. Move a portion of the SecuritySampleServer configuration to a shared configuration file

Tasks

1. Move the TCP/IP port definition settings from the primary SecuritySampleServer configuration file (`server.xml`) to a new, runtime-level shared configuration file named `global.xml`.

   a. Create text file `global.xml` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/shared/config`

      Then move the HTTP endpoint definition from `server.xml` to `global.xml`.

      Be sure to include `<server>` and `</server>` tags in `global.xml`, for example,

      ```
      <server>
        <httpEndpoint id="defaultHttpEndpoint" host="localhost"
        httpPort="${default.http.port}" httpsPort="${default.https.port}"/>
      </server>
      ```

   b. Within `server.xml`, add an include statement that points to `global.xml`, for example,

      Windows: `<include location="${shared.config.dir}\global.xml"/>`

      UNIX/Linux: `<include location="${shared.config.dir}/global.xml"/>`

      Be sure to place the include element between the `<server>` and `</server>` tags.

      Notice that the location attribute includes the variable shared.config.dir instead of a fully-qualified path to the shared configuration directory. This arrangement will allow the server to function properly if it is moved to a new base directory.

2. Review log messages related to the server configuration change.

   Use a text editor to view log file `console.log` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/logs`

   One message will indicate that `global.xml` was included in the server configuration. Another message will indicate that not functional changes were detected, given that the final server configuration is effectively the same.

3. Verify that the application can still be accessed using the new HTTP and HTTPS ports.
   a. Open a new web browser instance and visit `http://localhost:9081/SecuritySample`
   b. Log in as any one of the four users.
   c. Visit `https://localhost:9444/SecuritySample/ssl` (Accept the security exception.)
   d. Close the web browser.

F. Explore additional Liberty server command line interface options

Tasks

Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`
Then run the following commands:

a. Display and review the command line interface help.

```
server help
```

b. Display the version of the Liberty profile.

```
server version
```

c. Display the status of defaultServer and SecuritySampleServer.

```
server status defaultServer
server status SecuritySampleServer
```

G. Configure logging settings and dump problem determination to file

Tasks

1. Work with trace specification settings.

   a. Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`
   Then run the following command to stop SecuritySampleServer:

   ```
   server stop SecuritySampleServer
   ```

   b. Add the following property to file `bootstrap.properties` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`

   ```
   com.ibm.ws.logging.trace.specification="com.ibm.ws.webcontainer.*=fine"
   ```

   c. Open a command prompt and change to lab directory `WASv85Labs/LibCmdLine/wlp/bin`
   Then run the following command to stop SecuritySampleServer:

   ```
   server start SecuritySampleServer
   ```

   d. Using a new web browser instance, visit `http://localhost:9081/SecuritySample` and attempt to log in with invalid user name **charles** and password **charles1**. Then close the web browser.

e. Using a text editor, view the `trace.log` file in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/logs`

   Notice that the `trace.log` file includes numerous messages related the web container, but no messages related to security.

f. Add the following entry to the `server.xml` file in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`

   ```
   <logging traceSpecification="com.ibm.ws.security.*=fine"/>
   ```

   Be sure to place the `logging` element between the `<server>` and `</server>` tags.

   This trace specification entry will override the corresponding entry in the server `bootstrap.properties` file.

g. Using a new web browser instance, visit `http://localhost:9081/SecuritySample` and attempt to log in with invalid user name **susan** and password **susan1**. Then close the web browser.

h. View the `trace.log` file for server SecuritySampleServer. Notice that it now includes numerous messages related to security, but no additional messages related to the web container.

2. Work with the console log level setting.

   a. Using a text editor, view log file `console.log` in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer/logs`

      Notice that the `console.log` file includes AUDIT level authentication failure messages for the previous login attempts by invalid users charles and susan.

   b. Within the `server.xml` file for SecuritySampleServer, change the console log level to WARNING by adding `consoleLogLevel="WARNING"` to the `logging` element, for example,

      ```
      <logging traceSpecification="com.ibm.ws.security.*=fine"
      consoleLogLevel="WARNING"/>
      ```

   c. Using a new web browser instance, visit `http://localhost:9081/SecuritySample` and attempt to log in with invalid user name **kyle** and password **kyle1**. Then close the web browser.

   d. View the `console.log` file for SecuritySampleServer. Notice that it does not include an authentication failure message for the login attempt by invalid user kyle.

   e. Within the `server.xml` file for SecuritySampleServer, change the console log level to AUDIT by specifying `consoleLogLevel="AUDIT"` in the `logging` element, for example,

      ```
      <logging traceSpecification="com.ibm.ws.security.*=fine"
      consoleLogLevel="AUDIT"/>
      ```

   f. Using a new web browser instance, visit `http://localhost:9081/SecuritySample` and attempt to log in with invalid user name **lisa** and password **lisa1**. Then close the web browser.

   g. View the `console.log` file for SecuritySampleServer. Notice that it includes an AUDIT level authentication failure message for the login attempt by invalid user lisa.

3. Dump server problem determination information to file.

   The server dump command is useful for problem diagnosis of a Liberty profile server because the result file contains server configuration, log information, and details of the deployed applications in the workarea directory. The command can be applied to either a running or a stopped server. For a running server, the following information is also included:

   - State of each OSGi bundle in the server
   - Wiring information for each OSGi bundle in the server
   - Component list managed by the Service Component Runtime (SCR)
   - Detailed information of each component from SCR
   - Configuration Administrative data of each OSGi bundle
   - Information of registered OSGi services
   - Runtime environment settings such as JVM, heap size, operating system, thread information, and network status

   Run the following commands from lab directory `WASv85Labs/LibCmdLine/wlp/bin`

   a. Dump problem determination information with the server running.

   ```
   server dump SecuritySampleServer --archive="DumpRunning.zip"
   ```

   b. Stop the server.

   ```
   server stop SecuritySampleServer
   ```

   c. Dump problem determination information with the server stopped.

   ```
   server dump SecuritySampleServer --archive="DumpStopped.zip"
   ```

   Compare the contents of the two dump files located in lab directory `WASv85Labs/LibCmdLine/wlp/usr/servers/SecuritySampleServer`