Migrating from Apache Tomcat to IBM Bluemix

Lab Exercise

Contents

[Migrating Apache Tomcat to IBM Bluemix 6](#_Toc427178896)

[1.1 Sample application overview 6](#_Toc427178897)

[1.2 Testing the application on Tomcat before the migration 7](#_Toc427178898)

[1.3 Migrating the Easy JSP Forum application to Bluemix 18](#_Toc427178899)

[1.4 Deploying the application to Bluemix 26](#_Toc427178900)

[1.5 Testing the application on Bluemix 35](#_Toc427178901)

[1.6 Summary 36](#_Toc427178902)

[Appendix A. Notices Error! Bookmark not defined.](#_Toc427178903)

[Appendix B. Trademarks and copyrights Error! Bookmark not defined.](#_Toc427178904)

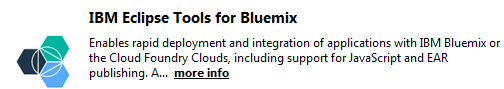
#### Overview

This Lab is designed to provide hands-on exercise on how to resolve incompatibilities in migrating an Apache Tomcat application to IBM Bluemix, Platform as a service cloud offering. IBM Bluemix allows developers and businesses to focus on their application development and data, while providing the platform to develop, run and manage web and mobile applications. Under the hood, IBM Bluemix uses Liberty server to run Java Applications.

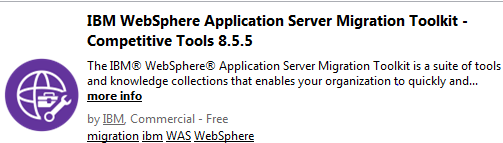
##### Pre-requisites

The following needs to be downloaded and installed before proceeding with the lab:

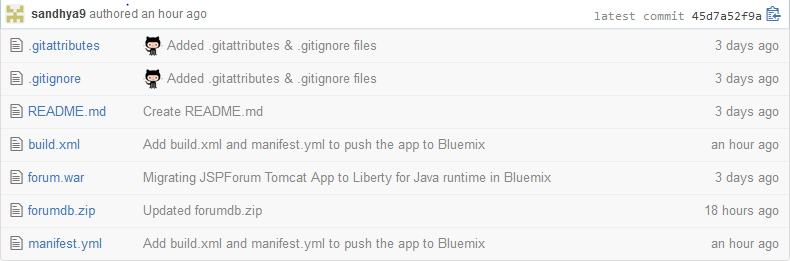
1. A valid installation of **JDK 1.7**
2. **Mars** (4.5) which can be downloaded from <https://www.eclipse.org/downloads/>
3. Install **IBM Eclipse Tools for Bluemix** from Eclipse Marketplace



1. Install **IBM WebSphere Application Server Migration Toolkit – Competitive Tools 8.5.5**



1. Install Binary Assessment Tool from : <https://developer.ibm.com/wasdev/downloads/#asset/tools-Migration_Toolkit_for_Application_Binaries_Tech_Preview>
2. Install Apache Tomcat 8.0 :
3. Download zip for Tomcat Application from <https://github.com/sandhya9/JSPForumMigration> and unzip to a folder on your system, we will call **<APP\_DIR>** in the Lab. Contents of the zip are as follows :



1. Download **Apache Derby database 10.11.1.1** from <http://db.apache.org/derby/releases/release-10.11.1.1.cgi>

Unzip to a folder <DERBY\_HOME>. Set environment variable DERBY\_INSTALL = <DERBY\_HOME>

# Migrating Apache Tomcat to IBM Bluemix

This lab exercise describes how to migrate a sample application from Apache Tomcat to Bluemix using the WebSphere Application Migration toolkit to facilitate and accelerate the process.

## Sample application overview

Easy JSP Forum is based on an open source project that resembles a forum where users can sign in, edit member information, post and reply to threads.

1. Prior to Migration: Tomcat App running on-premise using Derby DB
2. Post Migration to IBM Bluemix : App running in Cloud using Liberty for Java Runtime & SQLDB Cloud Service



Easy JSP Forum has been edited from the original application to connect to a Derby database using a JNDI data source and be distributed as a Web Archive (WAR).

## Testing the application on Tomcat before the migration

### Setup Tomcat 8.0 Server in Eclipse

* Create a Tomcat server under the **Servers** perspective in Eclipse.
* Right-click in the **Servers** view and select **New->Server**
* When prompted to select the server type, click **Apache ->Tomcat v8.0 Server**
* Click **Next**
* Click on **Download and Install** to install apache-tomcat binary. If you have already downloaded the binary , then point to the installation directory.
* Accept Apache License and scroll to location where Apache Tomcat is unzipped.
* From the list of Installed JREs, select a valid install of JDK 1.7.

Click **Next.** Then **Finish**.

Apache Tomcat v8.0 server is now running locally in Eclipse.

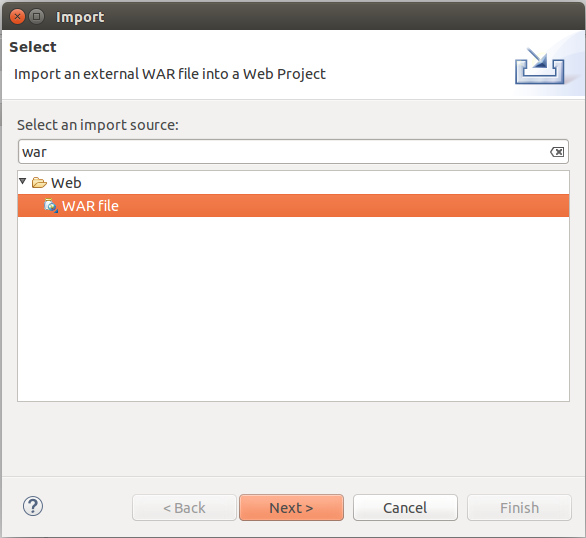
### Obtaining Easy JSP Forum application source code

We will import forum.war into Eclipse as a web project.

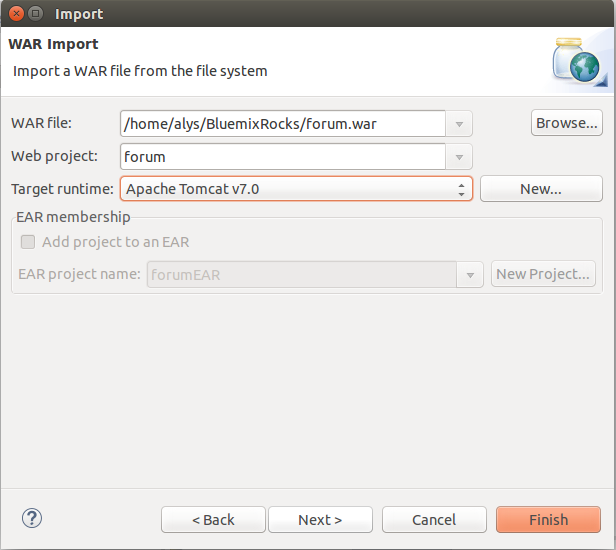
### Importing the Easy JSP Forum application into Eclipse

The Easy JSP Forum is distributed as a WAR file. In the next steps, we are going to import the project to Eclipse workspace and get it ready to be run on Tomcat.

1. Open Eclipse
2. Click **File > Import >**
3. Under **Web**, choose **WAR file** and click **Next**

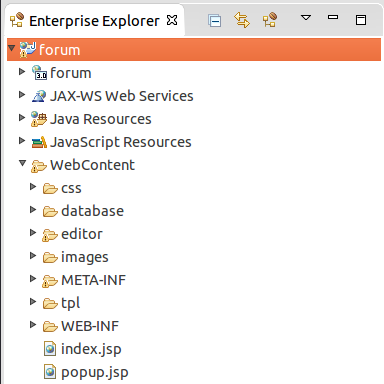


1. Click the **Browse…** button and browse to the **forum.war** file you downloaded
2. Remove the check on **Add project to an EAR**.



1. Click **Finish**

The application has now been successfully imported in Eclipse.



### Configuring the Derby database

To run the Easy JSP Forum application, you will need a data source to access the user and forum data displayed by the application. To keep things simple, this application will use Derby, a very lightweight and easy to use database.

Extract the compressed archive you downloaded for Derby 10.11.1.1 to a directory. This directory will be referred to as **<DERBY\_INSTALL>** folder.

The sample database forum used by the application has already been created for you. Simply extract the **forumdb** archive you downloaded and copy the extracted **forumdb** folder to **<DERBY\_INSTALL>/bin** folder.

### Start the Derby database

We are going to be using Derby as a Network Server.

1. Navigate to **<DERBY\_HOME>\bin** and run the startNetworkServer script

For Linux: startNetworkServer.sh

For Windows: startNetworkServer.bat

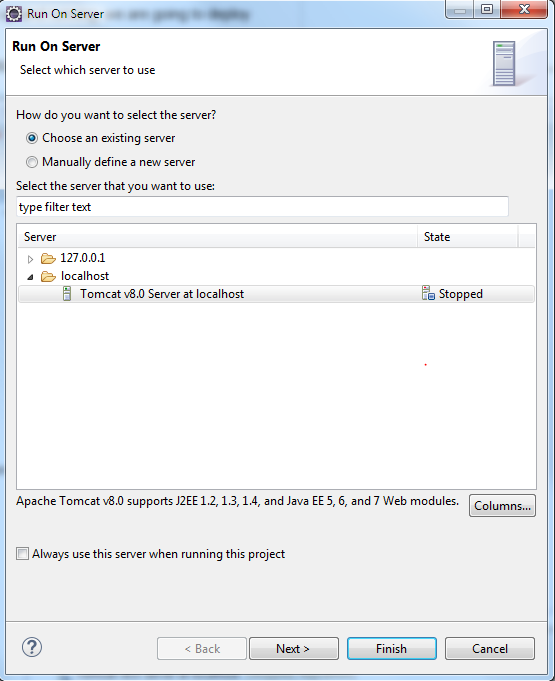
Derby database will start and display a message saying that Derby has been **started and ready to accept connections on port 1527**

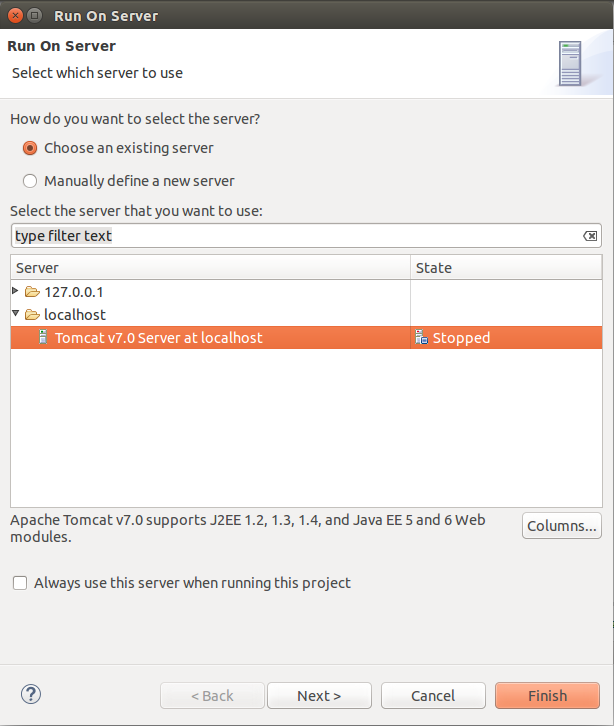


### Deploying the application in Tomcat

Now that the environment is ready and our database is up and running, we are going to deploy the application to Apache Tomcat from Eclipse:

1. Right-click the forum project in Eclipse and select **Run As > Run On Server....**
2. Choose the **Apache Tomcat Server** from the list

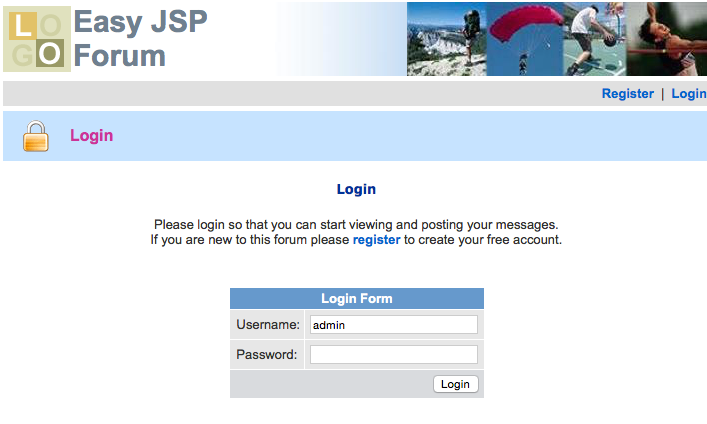




1. Click **Finish**

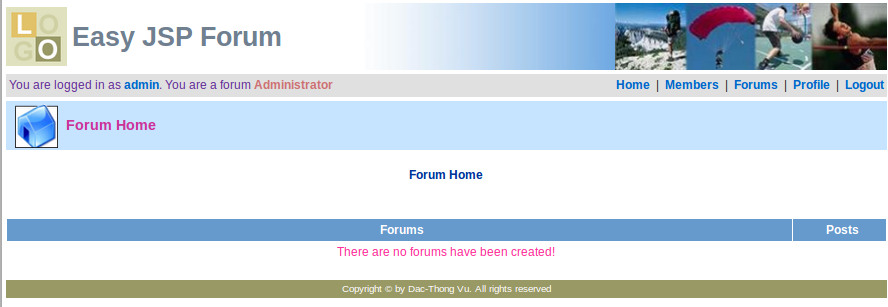
Tomcat will start, if it is not already started, deploy the application and open the browser to the welcome page URL e.g. <http://localhost:8080/forum>

**Note:** Port might vary depending from screen shot depending on the installation.



### Testing the application in Tomcat

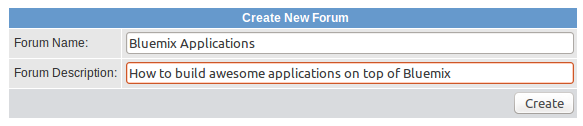
Login to the application using **admin** as Username and Password



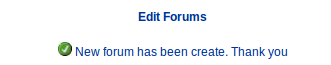
The screen shows the account page with the name of the user, **admin**, showing at the top.

Go to Forums and let’s create a forum. Go to **Forums** link on navigation bar at the top right part of the page.

In the **Edit Forums** page, enter a forum name and description and click **Create**



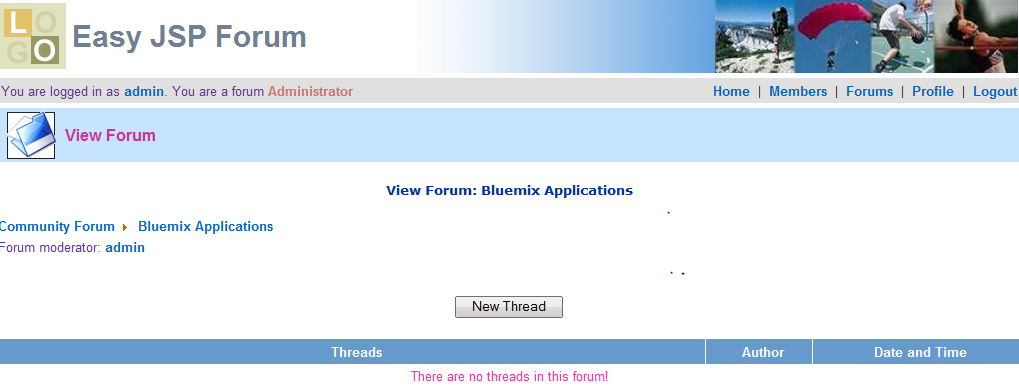
A message confirming the forum creation success is displayed.



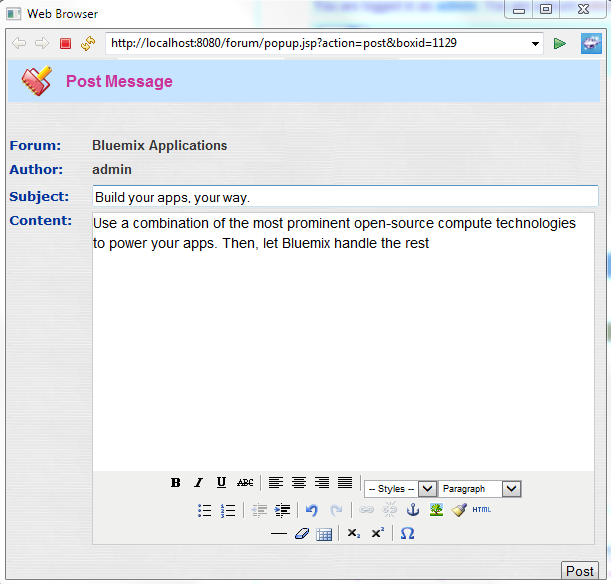
Please navigate to the **Home** page, you can see the newly created forum.



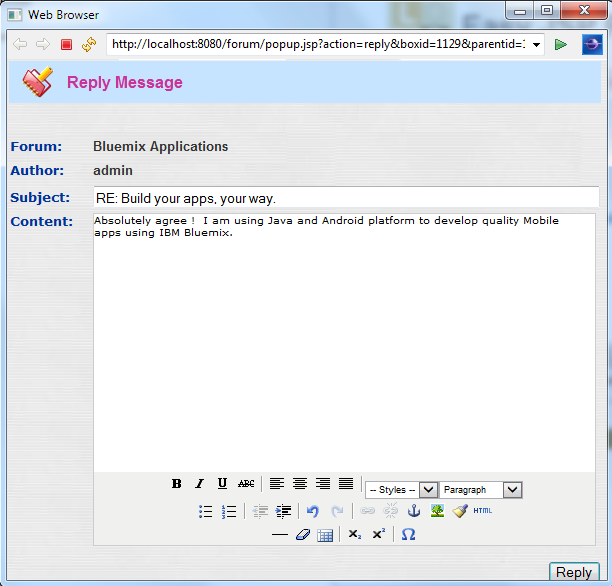
Play around with the application. Click on forum **Bluemix Applications** to start a **Thread** .



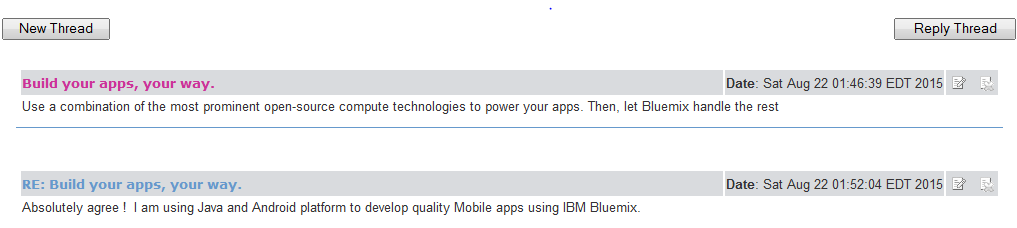
Click on **New Thread**



Ask a team member to click on Reply Thread and collaborate!



You can view the Forum Activity :



When you are satisfied, it’s time to start moving the application to Bluemix!.

You have successfully built, deployed, and tested Easy JSP Forum in Apache Tomcat 8. In the next session, we are going to start migrating the application to Bluemix.

## Migrating the Easy JSP Forum application to Bluemix

This section describes the necessary steps for migrating the application from Tomcat to Bluemix. Migration involves database creation on IBM Bluemix, Bluemix data source configuration, source code changes to connect to Bluemix data sources, and SQL syntax fixes.

The last part of this section describes how to deploy and test the migrated application.

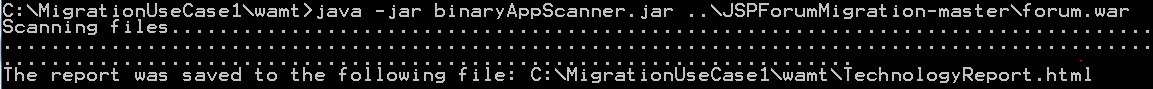
### Using the Binary Assessment Tool on Application Binaries

You have downloaded the tool as part of pre-requisites.

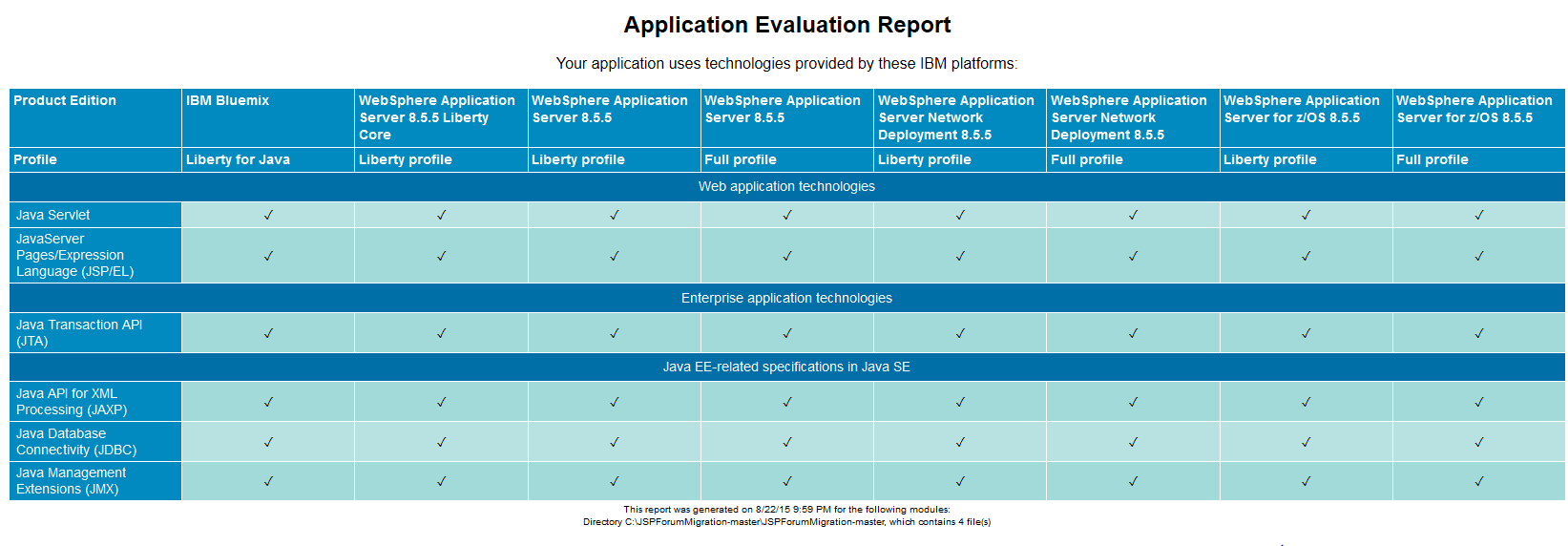
Install the tool : java -jar binaryAppScannerInstaller.jar

After installing, go to the wamt subdirectory

Run the analysis from the wamt subdirectory: java -jar binaryAppScanner.jar \directory-containing-war-file.



1. This command provides you with the high level technology analysis results (in HTML).



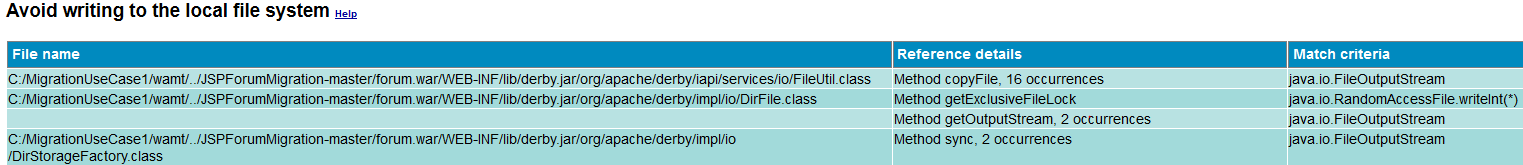
1. Next, you can run a detailed analysis report, which highlights potential issues as you move to the cloud. The files, line numbers, and includes reference material for handling these problems. From the wamt subdirectory: java -jar binaryAppScanner.jar \directory-containing-war-file --analyzeMigrationDetails --targetAppServer=bluemix. This also produces an HTML file.



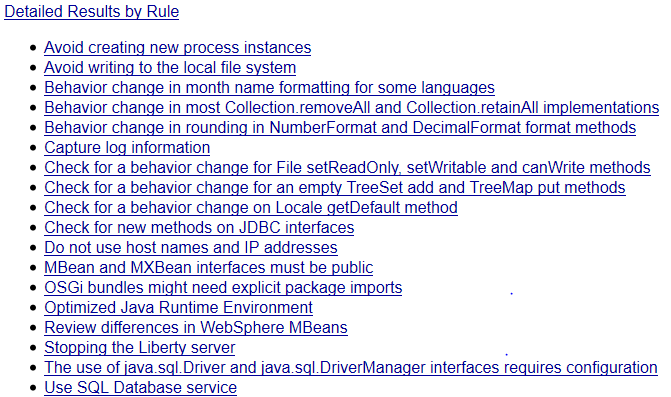
1. **Analyze results to choose your deployment topology**

The Tech preview of the binary scanner provides important analysis information for our application.

1. In the High Level Analysis, Under Web application technologies, we see that this application uses Servlet and JSP. This is, of course, not surprising for a Tomcat application. All the technologies will migrate to Liberty for Java runtime on Bluemix, as shown in column 1.
2. Under Java EE-related specifications, we see the application uses JDBC. This application will also run in Liberty for Java on IBM Bluemix, however, we will need to decide how to actually handle the database.
3. The detailed analysis provides us even more information. This report may be a little scary when you first look at it. Fortunately, most of the areas flagged are from the local Derby database we used with Tomcat, and don't apply when we move to the cloud. You can easily see the rules that are flagged from derby files by looking at the File name detail.



1. Once we eliminate the issues coming from the files with derby, the rules that we need to pay attention to for cloud migration are Capture log information, Stopping Liberty Server, and Use SQL Database Service.



1. For the database, the recommendation is to use SQL Database service in Bluemix, we will do that in this Lab.
2. The report also flags Capture Log Information as something to check for in the application. When we look at the help, we see: "Applications can log to System.err/System.out so that Cloud Foundry's Loggregator component picks up the log entries. Using Loggregator, you can tail application logs and dump a recent set of application logs." In our application, we're using the standard Java logging facility and also logging to the default location, which is System.out. So it will be automatically picked up by the Loggregator.



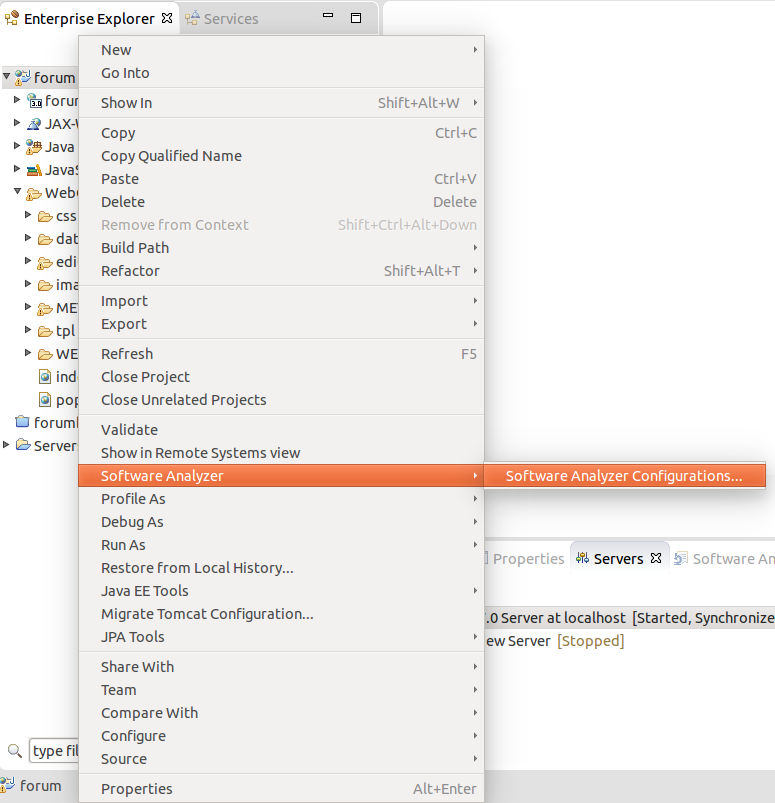
### Using Migration Toolkit for Competitive Migration

Migration Toolkit for Competitive Migration is provided by IBM to assist migrations from third party application servers to WebSphere Application Server, Liberty and Bluemix. The Migration Toolkit accelerates migrations, makes them quick and cost efficient.

The Migration Toolkit scans applications and reports any potential migration problems. The tool has contextual help that explains the finding and provides suggestions on how to fix it. For some cases, quick fixes exist that can apply changes to the code on your behalf.

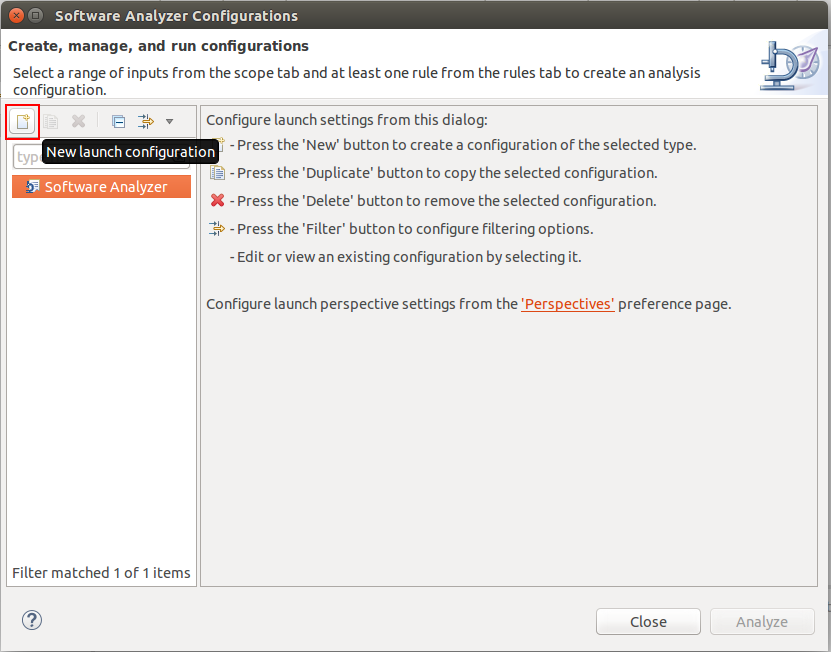
**Note:** Make sure you are using the **Java EE** perspective.

1. Since you have already installed Competitive Migration Toolkit, right-click the forum project in the Enterprise Explorer and select **Software Analyzer > Software Analyzer Configurations....**

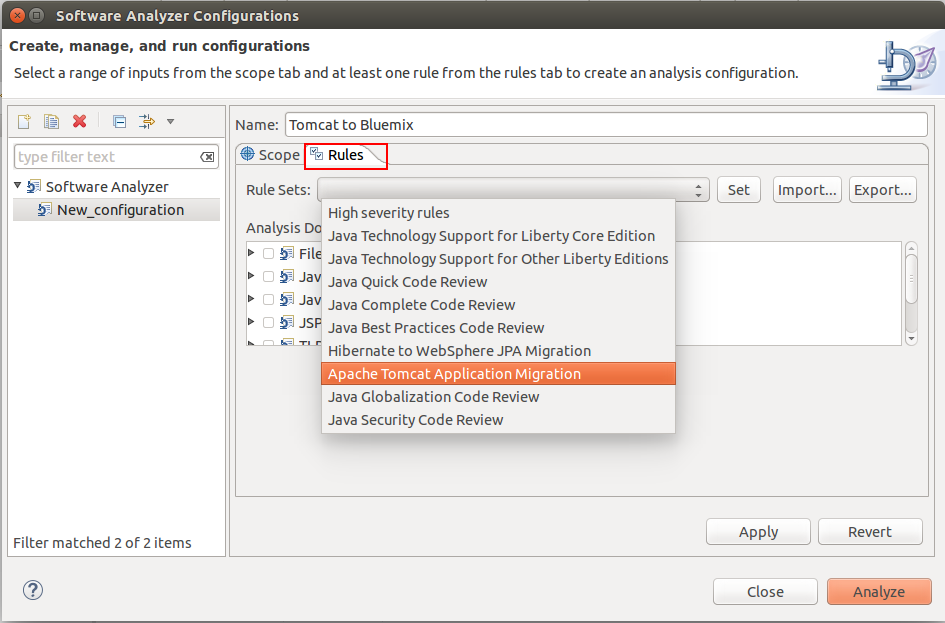


1. Click **Software Analyzer** in the left pane and then click New launch Configuration icon to create a Software Analyzer configuration

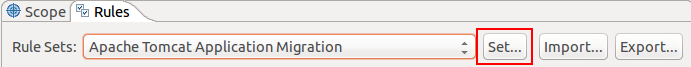
Note: Alternatively, you can double click Software Analyzer in the left pane to the same effect



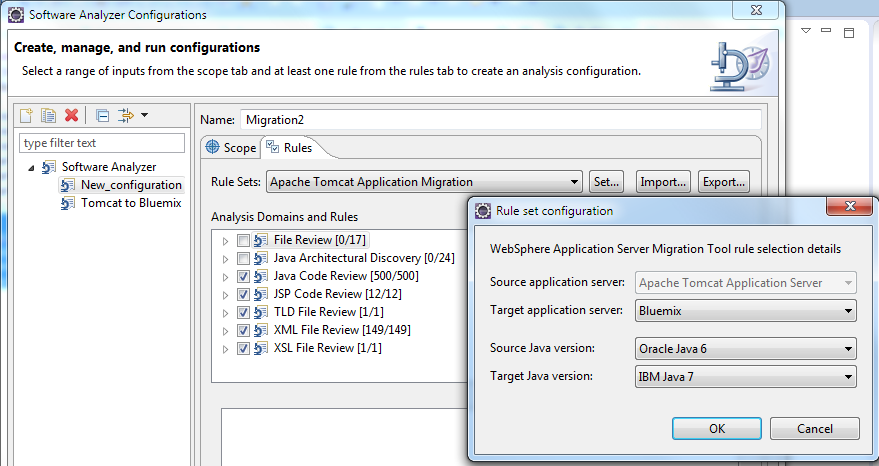
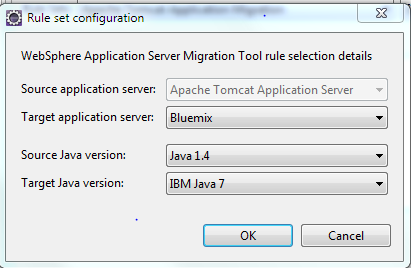
1. Enter a name for the configuration, such as **Tomcat to Bluemix**
2. Click on the **Rules** tab



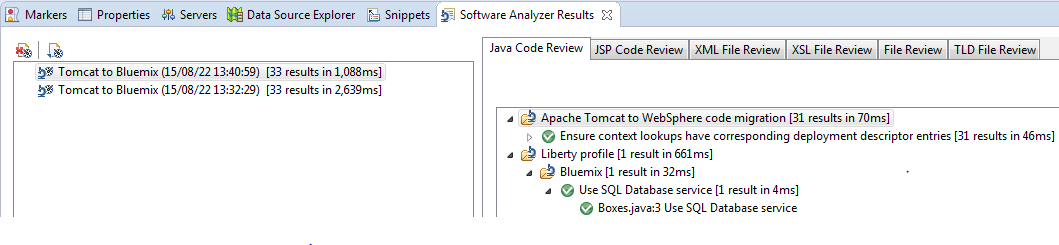
1. Click the drop down menu for the **Rules Sets**, choose **Apache Tomcat Application Migration** and click the **Set…** button



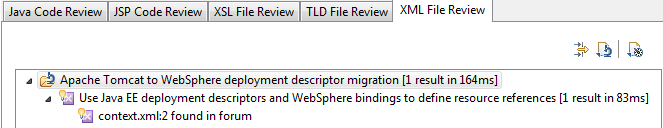
1. In the **Rule set configuration** dialog box, set the Target application server to **Bluemix**, the Source Java version to **Oracle Java 6**, and the Target Java version is **IBM Java 7**, and click **OK**

1. Click **Apply**
2. Click **Analyze** to start scanning the code and display the results in the **Software Analyzer Result** view



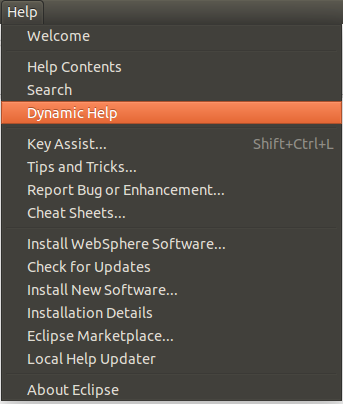
Output from XML Code Review is as follows :



The results of the analysis report shows no problems in JSP, XSL or TLD files but it shows 31 results in Java code review and 1 result in XML code review.

To understand more about the issue, we are going to enable the contextual help;

1. From the **Help** menu, choose **Dynamic Help**

****

1. Click on the finding in **Java Code Review**, the Contextual help provided is “**Ensure context lookups have corresponding deployment descriptor entries”**

This rule flags initial context lookups, so you can make sure there are corresponding entries in web.xml. If we check web.xml,

there is a resource-ref element for the datasource, so we can safely ignore this finding.

<resource-ref>

<res-ref-name>jdbc/easy</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

<res-sharing-scope>Shareable</res-sharing-scope>

</resource-ref>

**Tip:** You can right click and choose ignore on each of the findings to prevent them from appearing in future runs. In this case, we will

just ignore the result.

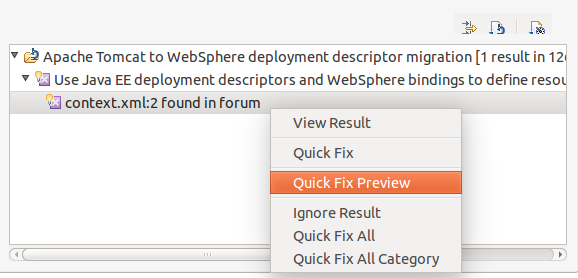
1. Click on the finding under **Liberty Profile, Bluemix**, which suggests **Use SQLDB service**
2. Click on the finding in **XML File Review**, the help provided is “**Use Java EE deployment descriptors and WebSphere bindings** **to define resources”**

This rule flags the resource information in META-INF/context.xml as they can be migrated to web.xml and WebSphere bindings. Notice

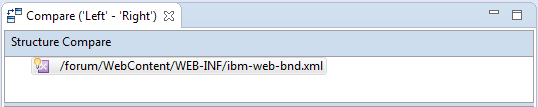
the little light bulb  next to the finding. This means a quick fix exists for this finding and Eclipse can make that change for you.

Let’s check the changes the toolkit is going to apply.

1. Right click the finding instance and choose **Quick Fix Preview**



Quick Fix Preview shows that it is going to make a change in **web.xml** and is going to create the **ibm-web-bnd.xml**.



1. We are going to allow the toolkit to make the changes. Right click the finding instance and click **Quick Fix All**

The application is ready to migrate to “Liberty for Java” runtime in Bluemix. We are going to remove the libraries in the WEB-INF\lib

folder since we don’t need them on the cloud and it is adding to your application size.

1. Go to WEB-INF\lib, choose the three derby jars and delete them

We have successfully prepared our application for the move to Bluemix. In the next section, we are going to deploy and start our

application on the cloud.

## Deploying the application to Bluemix

In this section you will use the Eclipse tools for IBM Bluemix to deploy the Easy JSP Forum application to IBM Bluemix, follow the steps below to do that.

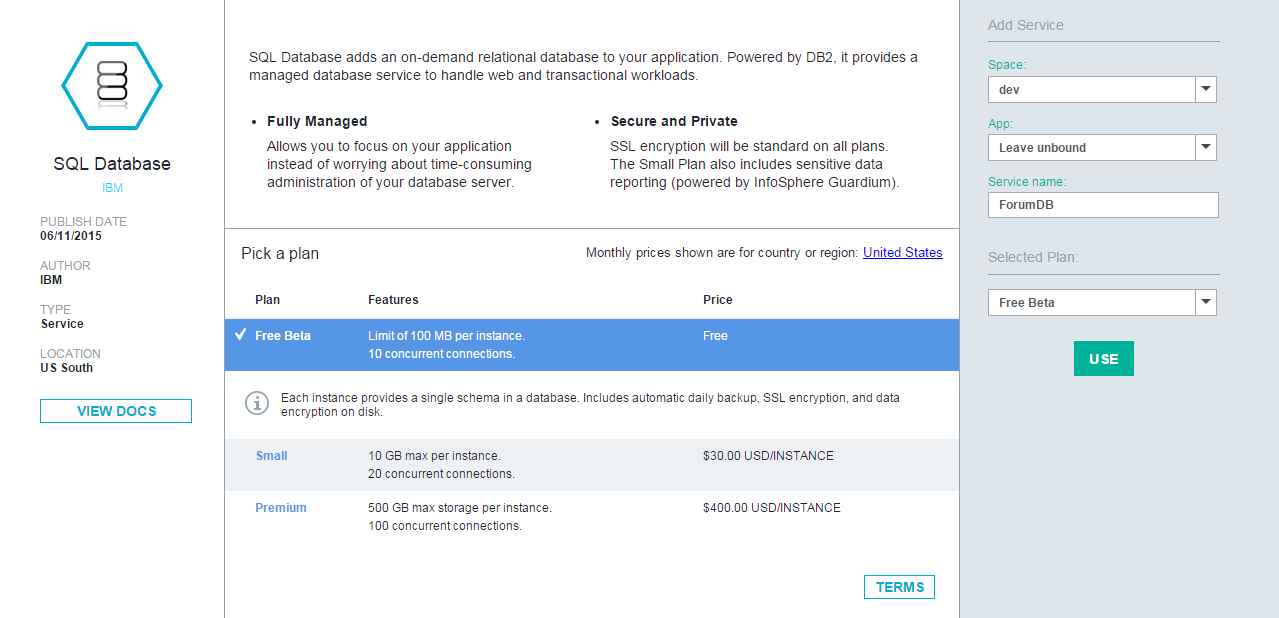
### Create the database on Bluemix

The Easy JSP Forum comes with Microsoft Access database. We have chosen SQL Database service in the Cloud as the database backend for the migrated application. SQL database in Bluemix, is an on demand relational database, powered by DB2. There is no database creation script that is provided with the application. In our scenario, we create the database and tables manually. Follow the steps below, to create the database and its tables

1. Login to Bluemix and open Bluemix Catalog
2. Search the catalog for the SQL Database Service provided by IBM and click it



1. The new SQL Database service dialog opens, name the service ForumDB, leave everything else to its defaults and click **USE**



1. After the Service is created, click **Launch** to open the SQL Database Web Console.
2. When the SQL Database Web Console opens, click “Work with Tables” To add the database tables, then click the + sign to add a table
3. Use the SQL statements below to create the needed database tables and populate it with sample data:

*CREATE TABLE BOXES ( "BOX\_NAME" VARCHAR(100) , "SORT\_DESC" VARCHAR(100) , "BOX\_ID" INTEGER , "MEMBER\_ID" INTEGER );*

*CREATE TABLE LEVELS ("GRADE" VARCHAR(100) ,"MIN\_POST" INTEGER ,"MAX\_POST" INTEGER ,"DESCRIPTION" VARCHAR(100) );*

*CREATE TABLE MEMBERS ("MEMBER\_ID" BIGINT NOT NULL , "USERNAME" VARCHAR(100) NOT NULL ,"PASSWORD" VARCHAR(100) NOT NULL ,"FIRSTNAME" VARCHAR(100) NOT NULL ,"LASTNAME" VARCHAR(100) NOT NULL , "EMAIL" VARCHAR(100) NOT NULL , "REGDATE" VARCHAR(100) NOT NULL , "TYPE" VARCHAR(100) NOT NULL );*

*CREATE TABLE "THREADS" ("THREAD\_ID" BIGINT NOT NULL , "BOX\_ID" BIGINT NOT NULL ,"PARENT\_ID" BIGINT NOT NULL , "MEMBER\_ID" BIGINT NOT NULL , "SUBJECT" VARCHAR(100) NOT NULL ,"POST\_TEXT" VARCHAR(1000) NOT NULL , "POST\_DATE" BIGINT NOT NULL );*

*INSERT INTO MEMBERS VALUES(1,'admin','admin','Forum','Admin','fxavier@br.ibm.com','Fri May 07 10:44:38 ICT 2010','Administrator');*

*INSERT INTO MEMBERS VALUES (2,'mod','mod','Forum','Moderator','dt@test.com','Fri May 07 10:44:38 ICT 2010','Moderator');*

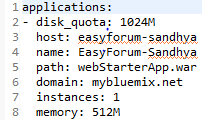
*INSERT INTO MEMBERS VALUES (3,'member','member','Forum','Member','dt@test.com','Fri May 07 10:44:38 ICT 2010','Member');*

The database setup is now complete

### Configuring the Easy JSP Forum application for Bluemix

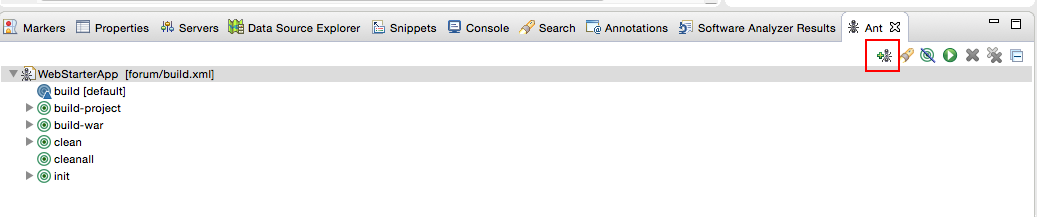
Now it is time to create the Bluemix application. You are going to use the Java Liberty Boilerplate to create and host your Bluemix application. Follow the steps below to push Easy JSP Forum application to IBM Cloud.

* 1. Copy build.xml and manifest.yml files from <APP\_DIR> folder to the root of **forum** project.
  2. The manifest.yml file is a configuration file used to instruct Bluemix on how to deploy your application, providing configuration for which application to deploy to, how much memory to use, etc. Inspect the file to see that it is configured to deploy your forum code to the Bluemix application you just created.

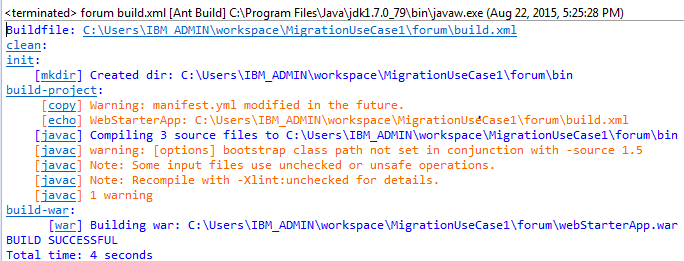


Pay attention to the hostname, it should be unique to deploy your application.

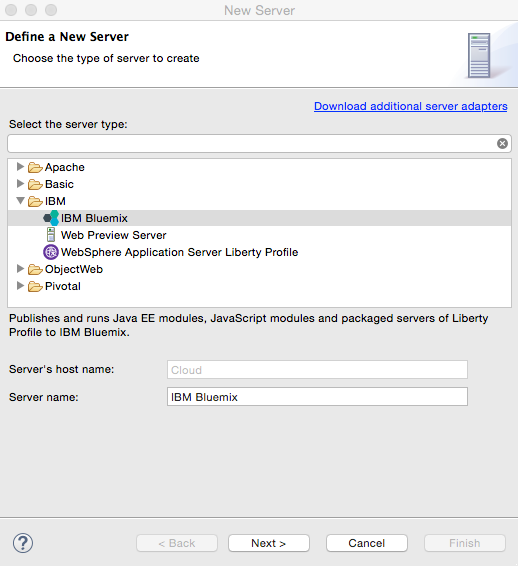
1. The build.xml is a small ant script that simplifies creating the WAR file from the project. Click Window -> Show View in Eclipse and select the ANT view.
2. From the Ant view click the add build file icon, and select build.xml from the forum project



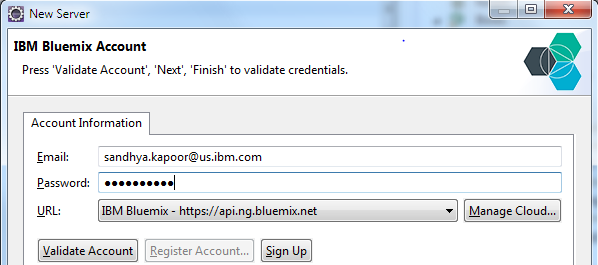
1. Double click the **build-war** target in ant view to execute building the WAR file. Wait until the build is successful then the war file should be generated in the forum root directory with the name webStarterApp.war



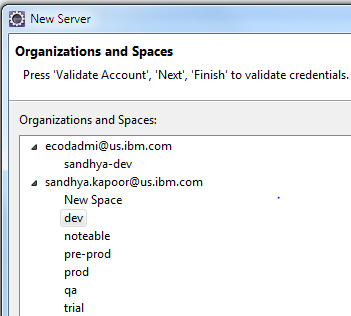
1. In Eclipse Java EE perspective, open the Servers view, right click on any free space and select New -> Server.
2. Create a new server, Go to **Servers** tab, Click **New-> Server**, select **IBM Bluemix** and click **Next**



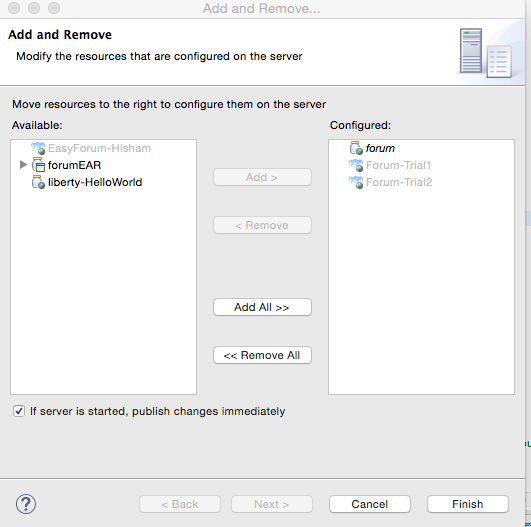
1. In the next view add your Bluemix user ID and password and click **Next**

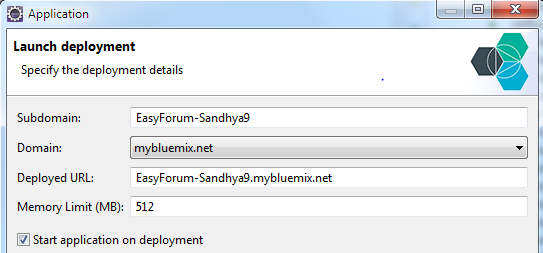


1. Select the organization and space (leave at dev for most users) and click Finish

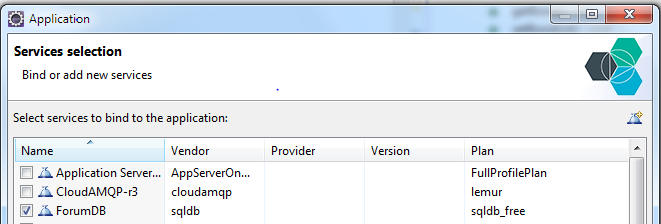


1. The IBM Bluemix server should be added to your servers view now. Right click the server and select **Add and Remove**
2. Highlight forum project on the left and click **Add**. Then click **Finish**

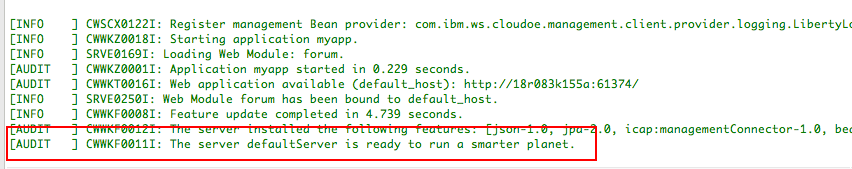




1. The add project wizard opens; click **Next** a couple of times until the Service Selection dialog. Select **ForumDB** to bind to the existing SQL Database service and click **Finish**.



1. The app will be pushed to Bluemix. Wait until it is complete. You should see a message like below



1. The application is now deployed to Bluemix and ready to be tested.
2. Liberty for Java Buildpack includes capability to automatically configure bound services. This makes porting an application to the Liberty for Java Buildpack on Bluemix much easier.

We are setting up our application to take advantage of the automatic configuration capabilities. JEE externalizes that kind of detail into an external application server configuration file (such as the server.xml that is used in WebSphere Liberty Profile) so that the program code only has to have an abstract reference to the service through JNDI. So, for instance, let’s say that we have a Servlet that wants to access JDBC DataSource. In JEE that’s commonly done through the use of a Resource annotation of the form:

*@Resource(name = “jdbc/library\_db”)*

In classes that are not JEE objects (e.g. Servlets or EJBs) you can use standard JNDI lookups to obtain references:

*(DataSource) ic.lookup("java:comp/env/jdbc/library\_db”);*

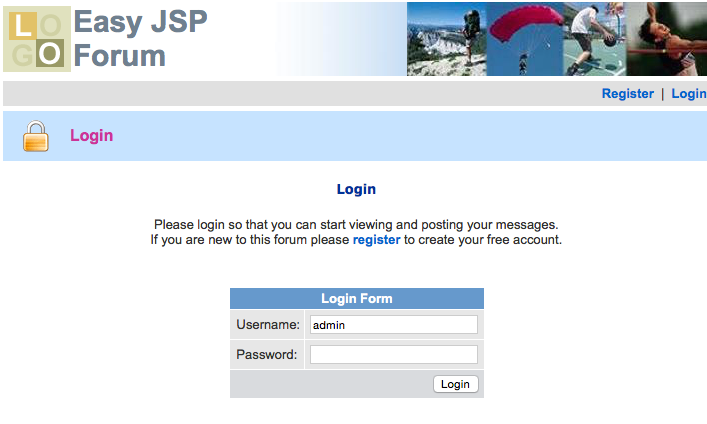
Now, where this meets up with the Liberty Profile is that the Liberty Profile Buildpack for Bluemix automatically configures these by writing a server.xml for the Liberty server,that contains the appropriate stanzas to do the binding. It will automatically create a JNDI binding of the form:

*<type, e.g. “jdbc">/service\_name*

for each service that is bound to your Liberty application. Since we bound a SQL Database service instance named “ForumDB” into our application with the manifest file, the Liberty Buildpack will automatically bind that instance to any @Resource statement or JNDI lookup of the type above.

If you want to see how Bluemix creates the server.xml for Liberty, you can see this for yourself by selecting a Java Liberty application deployed in Bluemix, selecting “Files and Logs” in the Dashboard, and then navigating down to app/wlp/usr/servers/defaultServer/server.xml

## Testing the application on Bluemix

1. Navigate to the application URL to test the app, the application URL should be <http://Easyforum-YOURNAME.mybluemix.net>/forum You should get the below home page
2. Use username admin and password admin to login to the Forum, test the application by creating some dummy forums and threads in these forums to test that the application on Bluemix is integrating with the SQL Database Service.



## Summary

In this exercise, you learned how to migrate an application running on Apache Tomcat 8 server to Liberty for Java runtime in Bluemix, after refactoring the application based on output from the Competitive Migration Toolkit. Tolearn more about the Migration Toolkit, please check the developerWorks article at <http://www.ibm.com/developerworks/websphere/library/techarticles/1404_vines1/1404_vines1.html>

We discussed refactoring the Tomcat application to be cloud ready, got the application running using Liberty for Java runtime in Bluemix and illustrated leveraging Cloud Services by binding the application to use SQLDB service in Bluemix.