

Using IBM DevOps Services & Bluemix Services

Part 4: Using IBM DevOps Services with RTC

This series of labs demonstrates how easy it is to use IBM DevOps Services and Bluemix together to plan, code, and deploy applications. Each part of the workshop guides you through the steps to experiment with key DevOps Services features

- **Part 1:** Deploy and make change to a simple application.
- **Part 2:** Add a data management Bluemix service and enable automatic deployment.
- **Part 3:** Use the planning capabilities to add a text messaging service, and enable traceability between code changes and work items.
- **Part 4:** Leverage automated builds and deployments, then explore integration between DevOps Services and Rational Team Concert (RTC)

After completing each part of the workshop, as a bonus, you'll also get applications that you can use to demonstrate DevOps Services and Bluemix capabilities to your colleagues and friends.

Before we get started, let's discuss the products you will be using in this workshop. **IBM DevOps Services** is a place where you can collaborate with others to develop, track, plan, build, and deploy software. **Bluemix** is an open-standards cloud-based platform where you can run the applications that you develop in DevOps Services. Bluemix provides a variety of services that you can leverage in your applications, so you don't have to start coding from scratch.

Learning Objectives

In this lab, you will learn how to do the following:

- Fork an existing application.
- Use the Pipeline Deliver capability for continuous delivery
- Connect to DevOps Services from an RTC Eclipse client.
- Triggers builds and deploymentst to Bluemix from the RTC Eclipse client.

Time Required:

This lab takes about 30 minutes to complete.

Before you begin

It is not required to complete labs 1, 2, and 3 before you start this lab. Nevertheless,

What do you need to begin this lab? You need to register at [Bluemix](#) and [DevOps Services](#). See the prerequisites for the worksop on <http://bluelabs.mybluemix.net> .

You'll also need one of the following modern browsers:

- Firefox 15 / Chrome 21 / Internet Explorer 10 / Safari 7 or later versions of thses browser

As indicated in the prerequisites for this workshop, you need to have Rational Team Concert V5 installed in order to complete this lab.

About the App

In this lab, you'll be working with a simple application called "Word Counter". This application, written in Java, uses the Jazz Source Control Management (Jazz SCM for short). Given some text, the application counts the number of words in the text.

Simple Word Counter

Please enter some text in the box below and press the submit button:

1- Fork the WordCounter application

To develop your own WordCounter application quickly, you want to leverage existing code. IBM DevOps Services provide this option with the fork capability. Forking a project means to take a copy of the code in an existing project and put it into a new project. Then you just have to modify and enrich the code with your own features.

1. Navigate to <https://hub.jazz.net/project/jlmarech/WordCounterDemo>
2. Click the **EDIT CODE** button in the upper-right corner. You are asked to sign in on JazzHub. Then the DevOps Services web development environment opens.
3. From the left pane, click the README.md file and read the application description. You can access other files to review the code of this application.
4. In the upper-left corner, note that at this point, the project belongs to another user (jlmarech) and is prefixed by the user name (jlmarech | WordCounter Demo)
5. Click the **FORK** button in the upper-left corner.
6. Type a name for your project.
7. Make sure that the options for **Scrum development** and **Bluemix deployment** are checked (see image below)
8. Then click **Save**.

Save a copy of the code into a new JazzHub project

Name your project: jlmarech | WordCounterDemo ✓

URL: http://hub.jazz.net/project/jlmarech/WordCounterDemo

Make it private (not public)

Add features for Scrum development ⓘ

Deploy to Bluemix ⓘ

Please select an organization and space to bill your services to:

Bluemix Organization: jlmarech@ca.ibm.com ▼

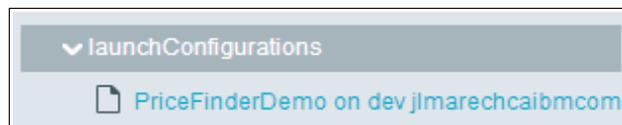
Bluemix Space: dev ▼

These selections can be changed later in the options for your Project Settings.

Save Cancel

When you fork a project, you create a copy of the source code in your environment: **<your-username> | <project-name>**

9. After the fork operation completes, verify if your project contains a **launchConfigurations** folder. If it does, delete the folder and the file it contains.



Now that you have a copy of the code in your own project, it is time to deploy and test it.

2- Create a pipeline

IBM DevOps Services provides a Pipeline Delivery capability for continuous software delivery. It enables teams to automate the build and deployment of their application, and to obtain early feedback on the new features they develop.

1. On the upper-right corner, click the **BUILD & DEPLOY** button, then click the **ADVANCED** button to access the Pipeline page
2. On the Pipeline page, click "**add a builder**". You will configure the automated builds for the WordCounter application



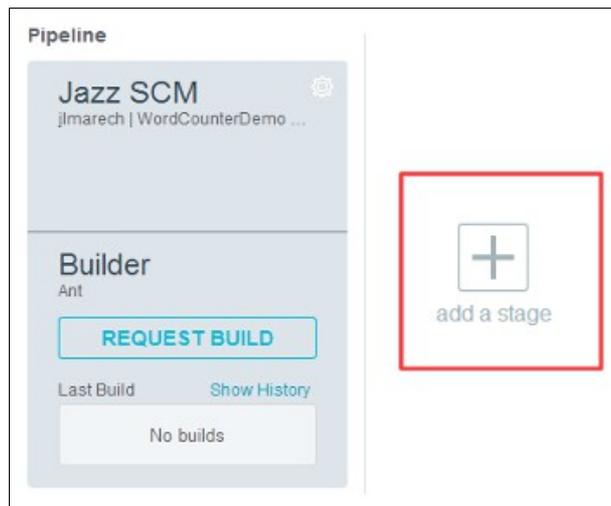
3. On the **Builder Configuration** page, enter the information shown in the picture below to configure your application builder. Builder, Stream, and Component are prepopulated. You need to:

- Enter your DevOps Services password
- Specify the build script path: `WordCounter`
- Specify the build archive directory: `WordCounterArchive`
- In the Build command section, modify the Ant command to:

```
ant -DARCHIVE_DIR=${ARCHIVE_DIR}
```
- Check the **Enable unit tests** option and verify the **Test result pattern** field contains `tests/TEST-*.xml`

Build script path	<input type="text" value="WordCounter"/>	i
Build archive directory	<input type="text" value="WordCounterArchive"/>	i
Build command	<pre>#!/bin/bash ant -DARCHIVE_DIR=\${ARCHIVE_DIR}</pre>	i
<input checked="" type="checkbox"/> Automatically build when a change is delivered/pushed		i
<input checked="" type="checkbox"/> Enable unit tests		i
Test result pattern	<input type="text" value="tests/TEST-*.xml"/>	i

4. Then click **SAVE** to create a builder.
5. On th Pipeline page, click on "**add a stage**" to configure autmatic deployment



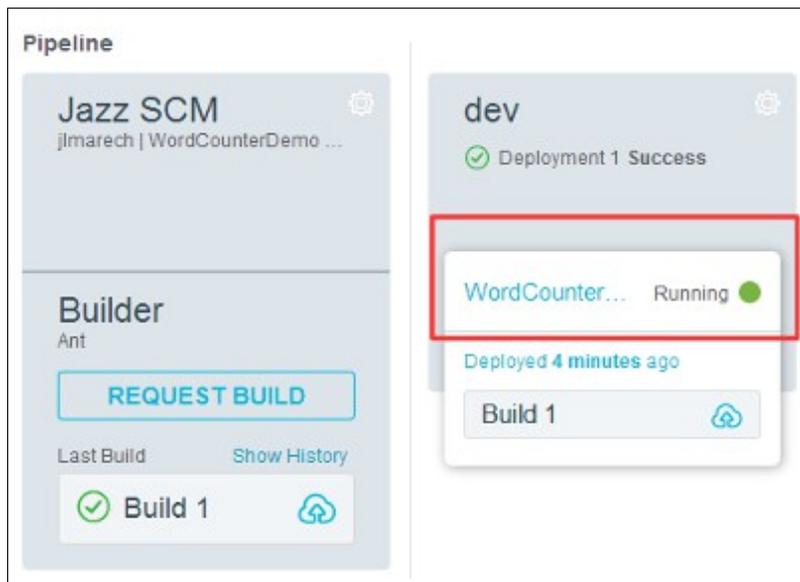
The deployer uses the information contains in the manifest file (manifest.yml). Here you will. In the configuration, you will just modify the script to create a unique host name for your application

6. In the script section, just after `cf push "${CF_APP}`, add the `-n` command line option followed by a unique host name, such as:
 - `-n <appnam>-<your_initials>-<some_timestamp>`

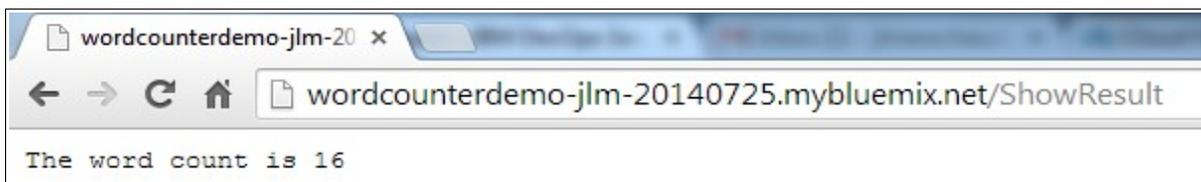
This command overrides the host name in the manifest file and ensure that you deploy your application under a unique hostname. In the example below, the command line option used is: `-n wordcounterdemo-jlm-20140725`

7. Then click **SAVE** to create the deployer.
8. Click the **REQUEST BUILD** button to build and deployed for the first time. Once the

build completes successfully (green check), then the deployment to Bluemix starts automatically. When the deployment is completed, the application is running (green circle) and the application is ready to be tested



9. Click the **WordCounter** link to access the home page of the application. You can type in some text and click Submit to verify that the application works properly.



3- See the pipeline delivery in action the application is updated

When a pipeline is defined, it can be used to requests build and deployment manually. But it can also be called automatically when a member of the team deliver a new change. In this section you will modify the code of the application, deliver a change, and then let DevOps Services build and deploy for you automatically.

1. In the DevOps Services project page, click the **EDIT CODE** button
2. On the left pane, under WordCounter / WebContent, select index.html to display its source code.
3. Change the text displayed on the home page by replacing `<h1>Simple Word Counter</h1>` with something else, for instance: `<h1>My Great New Word Counter</h1>`
4. On the left pane, click the Repository icon (Jazz SCM)



5. An unresolved changed is displayed. Click the **CHECK-IN** button.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Word Counter</title>
6 <link rel="stylesheet" type="text/css" href="style.css"></link>
7 </head>
8 <body>
9   <form action="ShowResult" method="POST">
10    <h1>Simple Word Counter</h1>
11    <h1>Simple Word Counter</h1>
12    <p>Please enter some text in the box below and press the submit button:</p>
13    <textarea rows="10" cols="80" name="content" class="content"></textarea>
14    <div><input type="submit" value="Submit"></input></div>
15  </form>
16 </body>
17 </html>
```

6. Under the **Outgoing Change Sets** section, enter a comment and click the **DELIVER** button. As the change will be delivered to the stream, it will trigger the automated build and deploy process.

Outgoing Change Sets DELIVER MORE

Workitem: Assign a workitem... - Current

Author: Jean-Louis Marechaux Last modified: 7/25/2014 2:52:33 PM

[Open Change Set](#)

[WordCounter/WebContent/index.html \(Modified\)](#)

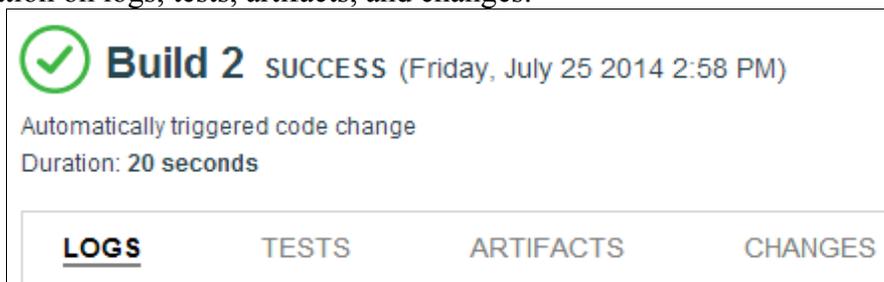
7. Go immediately to the **BUILD & DEPLOY** page (button in the upper-right corner). Notice that a new build has started automatically. When the build ends, the deployment is also started automatically.
8. When the deployment completes, click the **WorkCounter** link to open the home page. Notice that DevOps Services has built and deployed your application to include the latest changes that you have made.



4- Explore information on builds and deployments

The pipeline feature in DevOps Services provide a lot of information on builds and deployment. In this section, you will access the information pages and explore their content.

1. Go back to the DevOps Services Pipeline page.
2. Click on the build name (Build 2) under the **Latest Build** section.
3. The build information page is displayed, providing a summary and access to detailed information on logs, tests, artifacts, and changes.

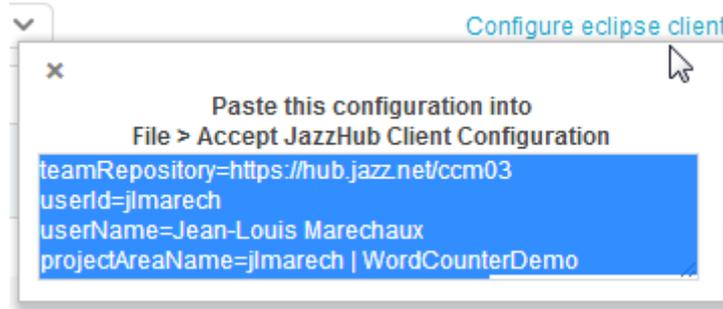


4. Review the information available for this build.
 - On the **LOGS** tab, review the different steps of the build. This information is useful to help you resolve problems when a build fails.
 - On the **TESTS** tab, click the **WordCountFinderTest** link to reveal details on unit tests performed during the build process
 - On the **ARTIFACTS** tab, explore the different artifacts that you can download.
 - On the **CHANGES** tab, review the changes associated to this build.
 - Notice that you also have access to all the other builds perform for this application
5. On the upper-left corner, click the **Pipeline** link to go back to the pipeline main page.
6. On the pipeline page, click on the grey area under "Deployment 2 Success". The deployment information page is displayed.
7. Review the detailed information regarding the deployment of your application to Bluemix. This information is useful to help you resolve problems when a deployment fails.

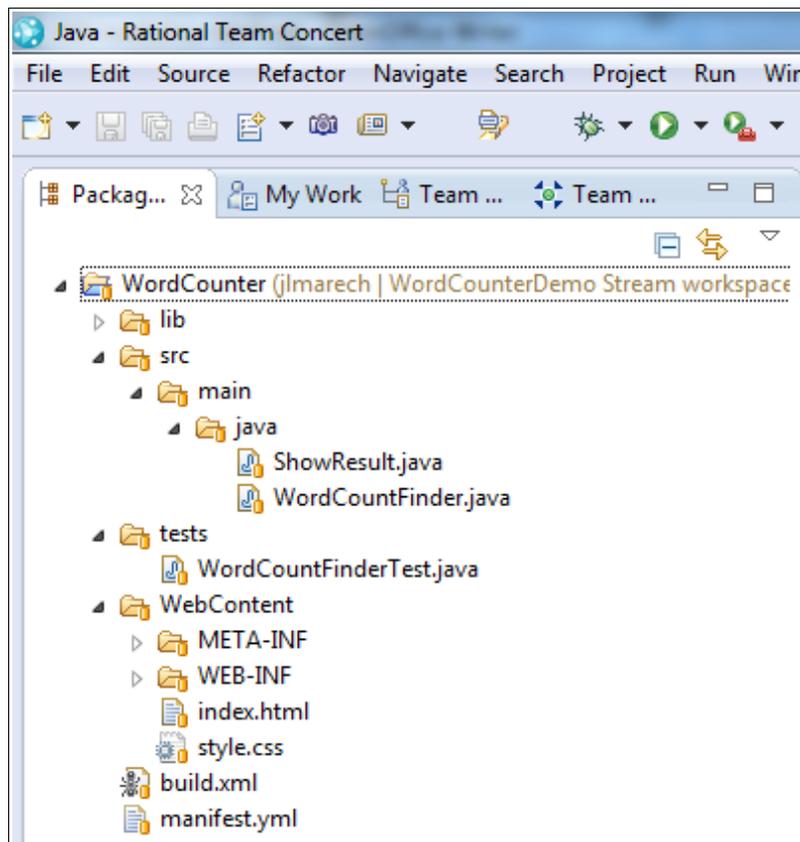
5- Work on DevOps Services from an Eclipse client

To use IBM DevOps Services, you can choose the web IDE and do all your work on the cloud. But you can also decide to leverage your favorite desktop client such as Eclipse or Microsoft Visual Studio IDE. Here you will use the Rational Team Concert for Eclipse to add a new feature to the WordCount application. In addition to the number of words, your application will also count and display the number of characters in the text entered by the user.

1. On the upper-left corner, click on the application name to go to the main DevOps Services page.
2. Click the **Configure eclipse client** link to display connection information



3. Copy the information. You will need it in Rational Team Concert.
4. Verify that Rational Team Concert for Eclipse is installed. Check the prerequisites for this workshop on <http://bluelabs.mybluemix.net>.
5. Launch Rational Team Concert and go to **File > Accept JazzHub Client Configuration**
6. Paste the information that you copied from DevOps Services and click **Next**.
7. When you are asked to login, enter the password from your IBM DevOps Services account and click **OK**.
8. On the "Create Repository Workspace" dialog, keep the default values and click **Finish**. Rational Team Concert (RTC) connects to DevOps Services on the cloud and downloads the code of your WordCounter application on your local hard drive.
9. On the RTC main menu, select **Window > Open Perspective > Java**. You now have access to the source code



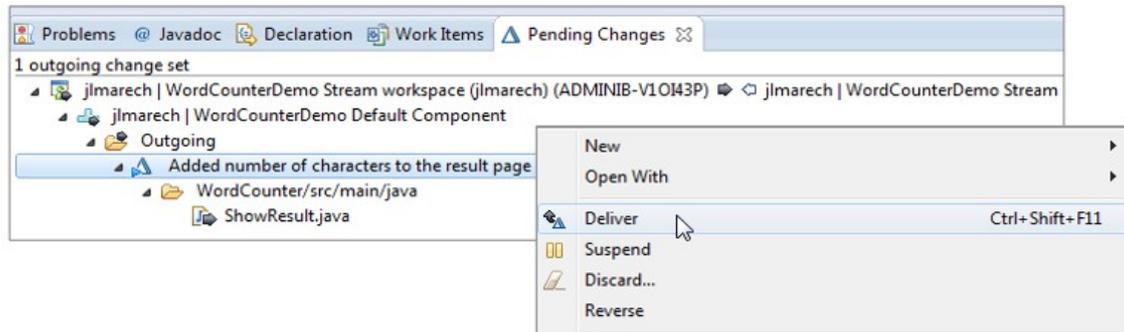
10. On the RTC main menu, select **Window > Show View > Pending Changes**. This view displays all the changes to be delivered to the source control management system.
11. From the left pane, open the **ShowResults.java** class. This is where you will modify the code to display the number of characters for the text submitted by the user.
12. In the **doPost** method, under the **Parse input** section, add:
`int charCount = wordCountFinder.countChars(content);`
13. In the **doPost** method, under the **Print response to the Web Page** section, add:
`out.println("The number of characters is " + charCount);`
14. Click **File > Save** (or Ctrl+S). The code is now similar to the one below (the new code is on line 36 and 42)

```

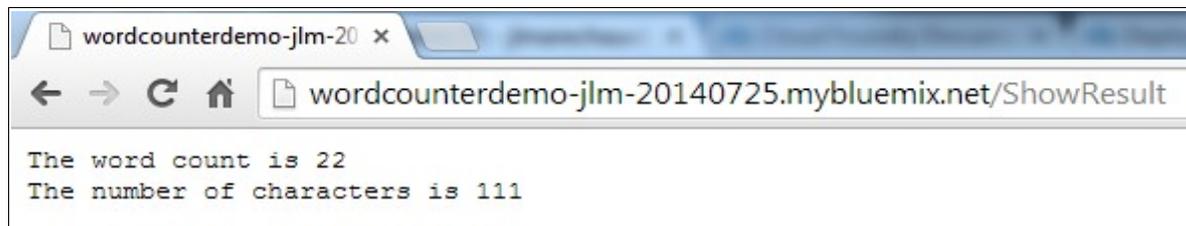
30 protected void doPost(HttpServletRequest request,
31     HttpServletResponse response) throws ServletException, IOException {
32
33     // Parse input
34     String content = request.getParameter("content");
35     int wordCount = wordCountFinder.countWords(content);
36     int charCount = wordCountFinder.countChars(content);
37
38     // Print response to the Web Page
39     response.setContentType("text/plain");
40     PrintWriter out = response.getWriter();
41     out.println("The word count is " + wordCount);
42     out.println("The number of characters is " + charCount);
43 }
44
45 }
46

```

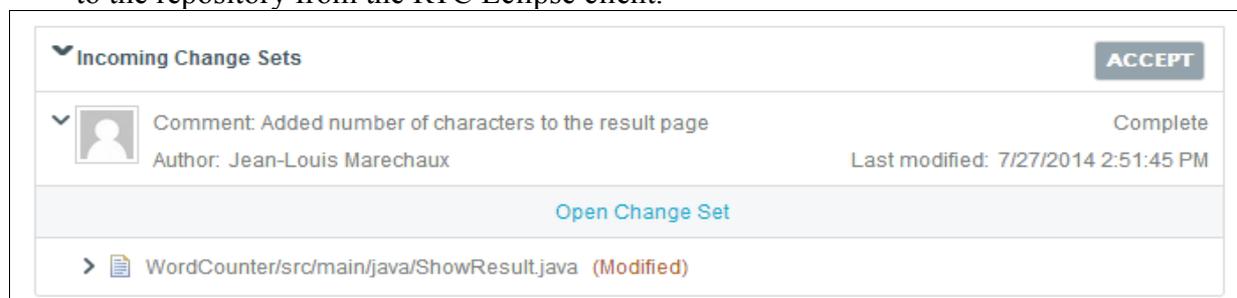
15. In the **Pending Changes** view, right-click the modified class and select **Check-In > New Change Set**. Your change is ready to be delivered.
16. In the **Pending Changes** view, under the **Outgoing** section, right-click `<Enter a comment>` and select **Edit Comment**. As a comment, type in: Added number of characters to the result page
17. Right-click the change set (where you added the comment) and select **Deliver**



18. The code is delivered to the main repository. Because you created a pipeline for the application, the code delivered from RTC triggers a build and a deployment.
19. In your web browser, go to the IBM DevOps Service page of your WebCounter application. Open the **BUILD & DEPLOY** page and notice that a new build has been triggered. When the build and the deployment complete, test your application. It now gives the number of characters for the text submitted.



20. In DevOps Services, go to the **EDIT CODE** page, then click the **Repository (Jazz SCM changes)** icon on the left pane.
21. Notice that there is an incoming change set. It contains the changes that you have delivered to the repository from the RTC Eclipse client.



22. To get the latest changes included in your view on DevOps Services, click the **ACCEPT** button. If you open the code from DevOps Services, you will now see the changes that you submitted from RTC.

Summary

In this lab, you learned how to:

- Fork an existing project (copy code for reuse)
- Create a pipeline for automated builds and deployments
- Modify source code from the web IDE
- Deliver code changes to the source control management system (Jazz SCM)
- Explore information provided on builds and deployments by the DevOps Services Pipeline Deliver capability.
- Modify code from the RTC Eclipse client, and deploy on Bluemix.

Next Steps...

This lab is the last part of the *"Using IBM DevOps Services & Bluemix Services"* workshop. You can find further information on DevOps Services and Bluemix from the workshop main page at <http://bluelabs.mybluemix.net>