Cloud Developer Certification Preparation

Exercise 5.3:

Debugging cloud applications by using IBM Bluemix DevOps Services web code editor

Exercise 5.3: Prerequisites

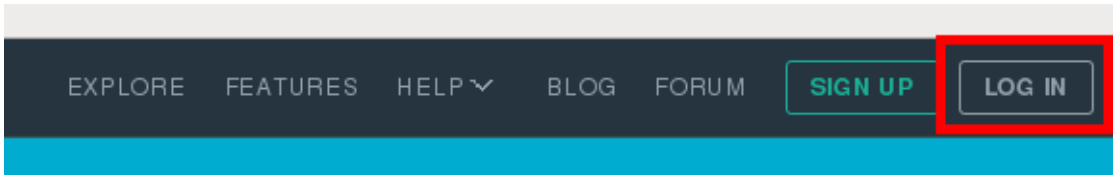
Sign up for a 30-day free trial IBM Bluemix account if you don't already have one. Also, create a DevOps account for sample code download: https://hub.jazz.net/. You should use the same credentials for both the Bluemix and DevOps accounts.

You also need the following software:

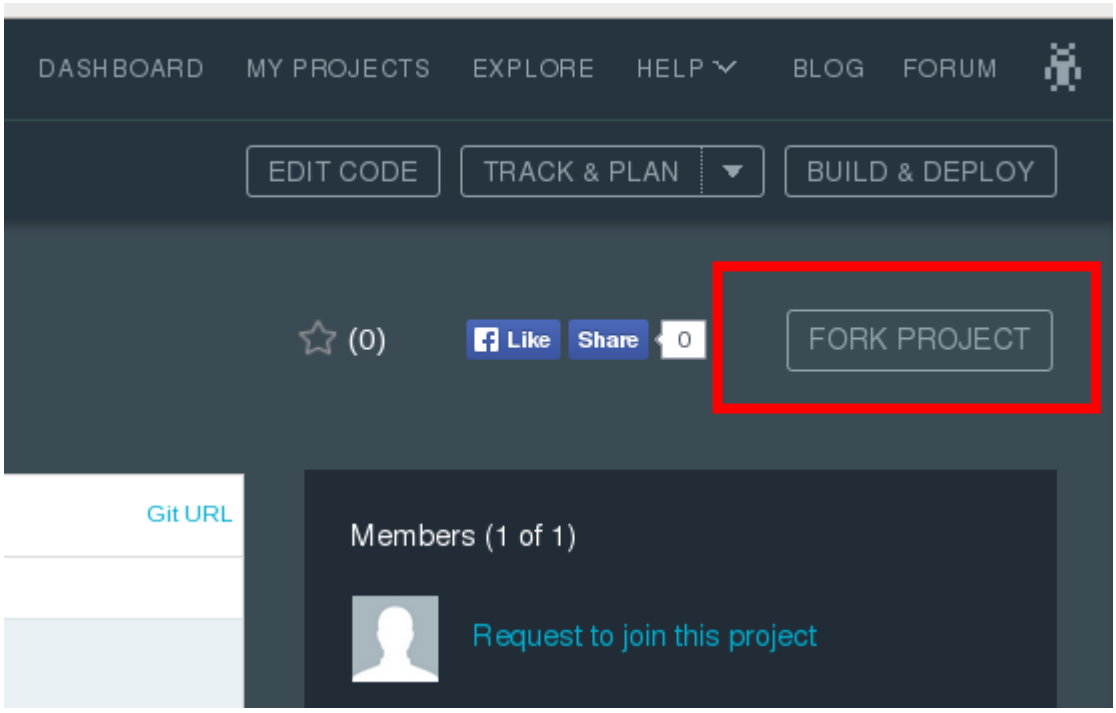
- A web browser supported by Bluemix:
    - Chrome: the latest version for your operating system
    - Firefox: the latest version for your operating system and ESR 31 or ESR 38
    - Internet Explorer: version 10 or 11
    - Safari: the latest version for the Mac

Exercise 5.3.1: Forking a project

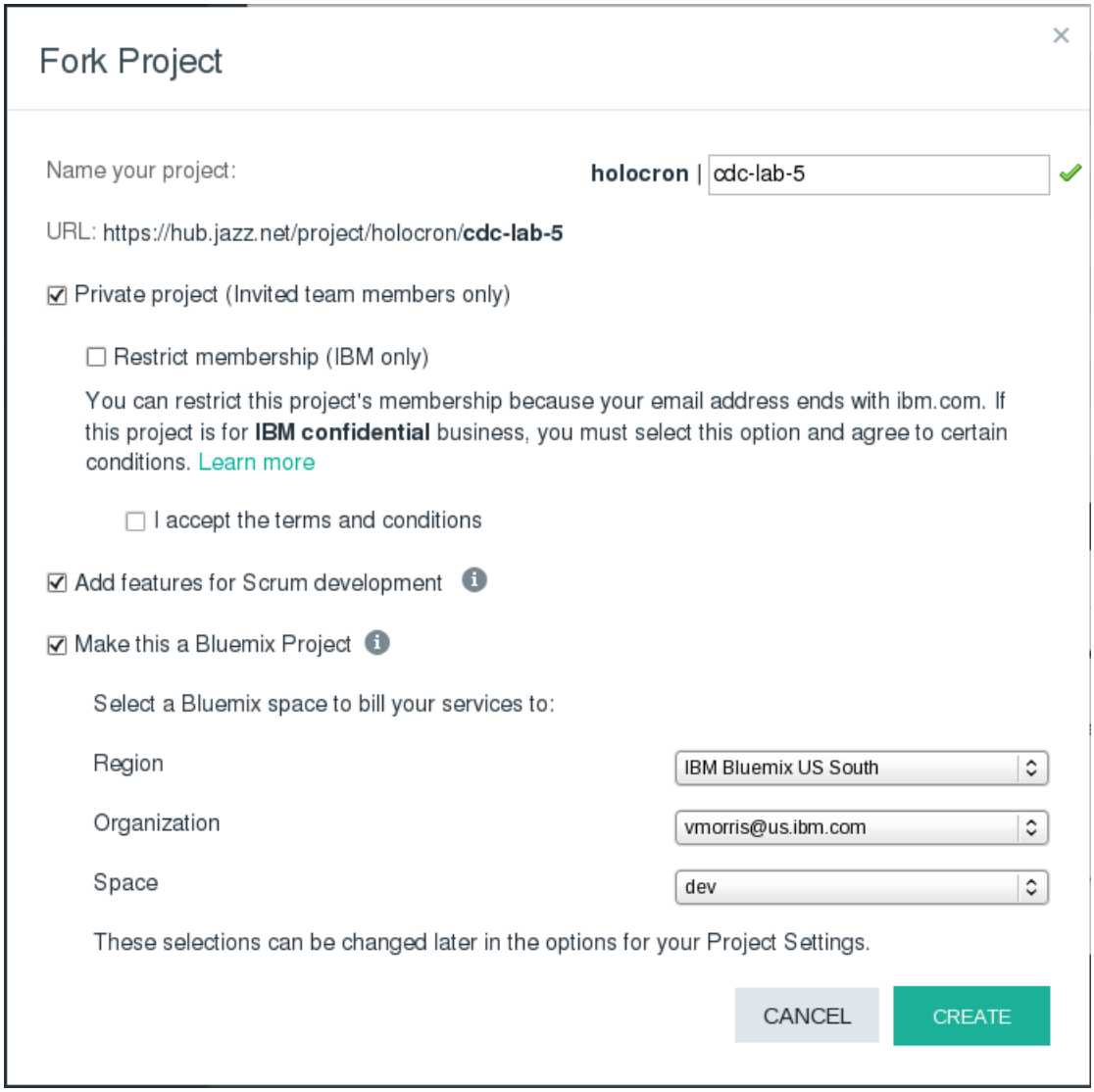
1. Fork the lab project in DevOps Services. Open your browser and navigate to http://hub.jazz.net. Then, log in.



2. Open the URL https://hub.jazz.net/project/ecosysdevcnc/cdc-lab-5/overview and click FORK PROJECT.

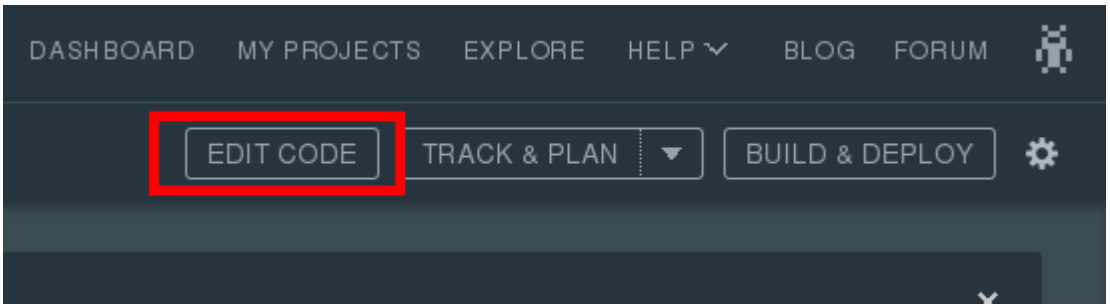


3. In the Fork Project dialog, enter a unique name for the project, and select the Private project, Add features for Scrum development, and Make this a Bluemix Project check boxes. Configure an appropriate Bluemix space and then click CREATE.

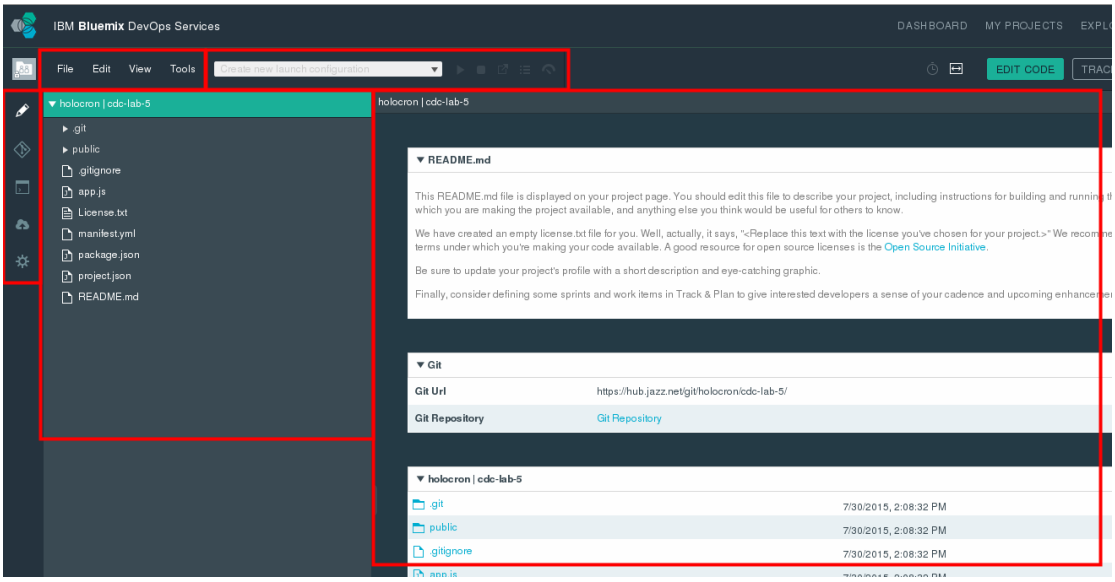


Exercise 5.3.2: Using the DevOps Service Web Code Editor

1. After you create the new project, click EDIT CODE.



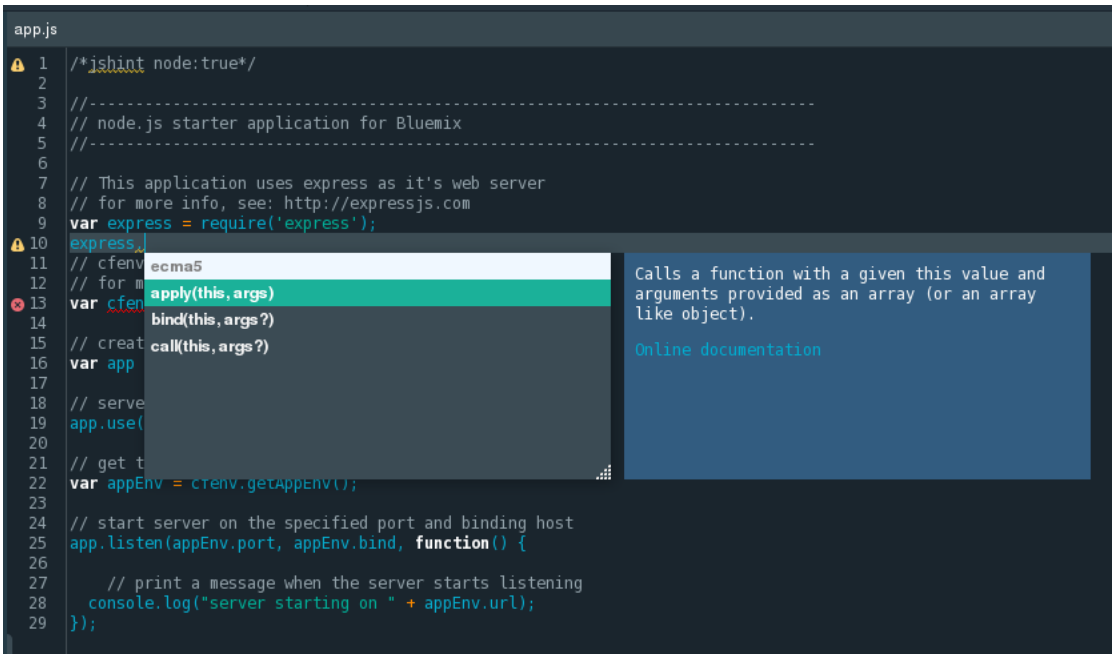
This is the DevOps Services web code editor. Along the top, you will find the menu bar, with options File, Edit, View, and Tools. To the right are the Bluemix runtime control buttons. Along the left column are various buttons for switching between the code editor, Git repository, shell, applications, and options views. Immediately to the right of these buttons is the file system explorer, and to the right, taking up the majority of the screen, is the code editor and preview pane.



2. Click **app.js** to open it in the editor and note how the editor automatically assigns colors based on the knowledge that the file contains JavaScript code.
3. On line 10, after the `var express` assignment, add the following text (including the period):

```
express.
```

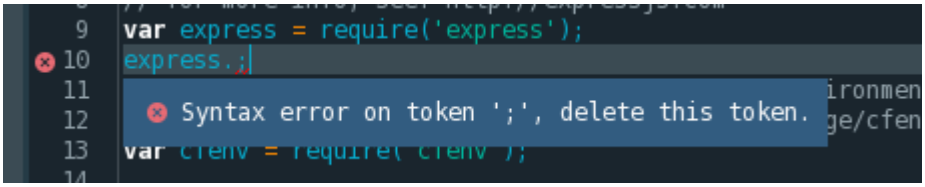
Note how the editor shows a dialog with suggestions for completing the statement and also includes text from and links to the online documentation for the particular API.



4. Without selecting any of the options presented, enter a single semicolon to finish the JavaScript statement.

```
express.;
```

Note how the editor intelligence marks the line as an error with a red X. Hovering the mouse pointer over the red X opens a dialog with details about the error.



Important: Be sure to delete this line before continuing with the remainder of this lab.

5. Select the public directory and then click File > New > File. Name the new file about.html. The editor automatically opens the new file for editing. Enter the following HTML code into the editor:

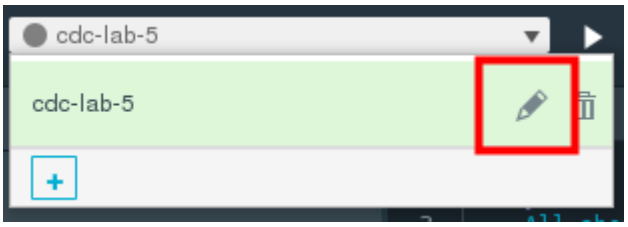
```
<html><body>   All about me!</body></html>
```

The editor automatically indents and highlights the code as you type it in.

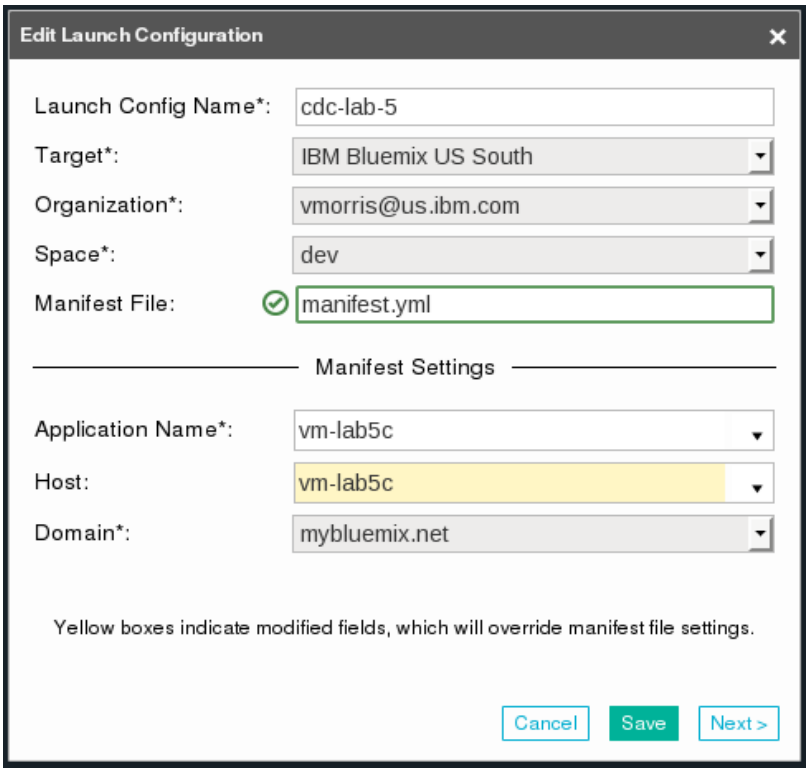
It is not necessary to save the file manually because the editor will automatically save your changes after every change. You can manually force a save of a file by clicking File > Save.

Exercise 5.3.3: Deploying an application to Bluemix from DevOps Services

1. Locate the drop-down dialog in the Bluemix runtime control area and click the down arrow. Then, click the Pencil icon next to your app/project name.

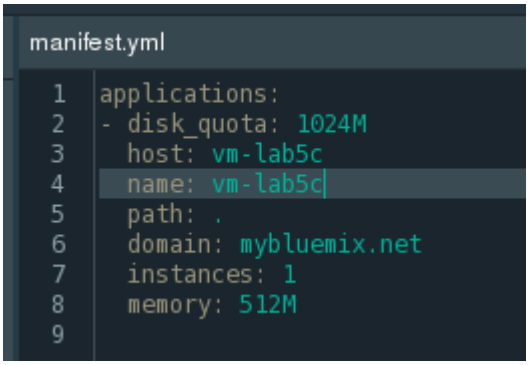


2. In the Edit Launch Configuration dialog, enter a unique name in the **Application Name** and **Host** fields. These names should match, and combined with the domain, will create the Bluemix route through which you can access your running application. Click **Save**.



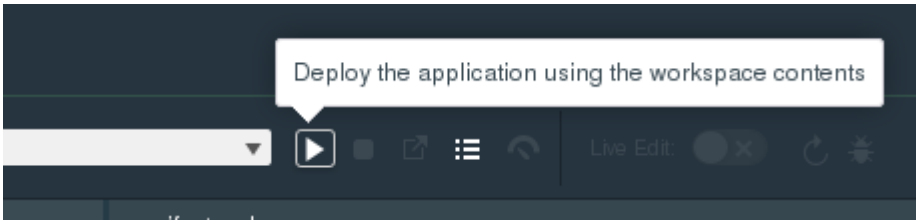
These configuration settings override the settings located in a file called manifest.yml.

3. Open the manifest file in the editor and overwrite the `host` and `name` parameters with the same unique name that you set in the launch configuration dialog.



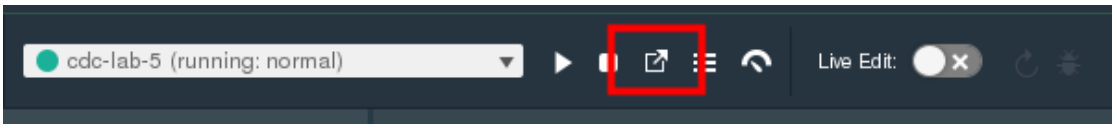
You are now ready to deploy the application to Bluemix.

4. Click the **Deploy** icon in the Bluemix control pane.



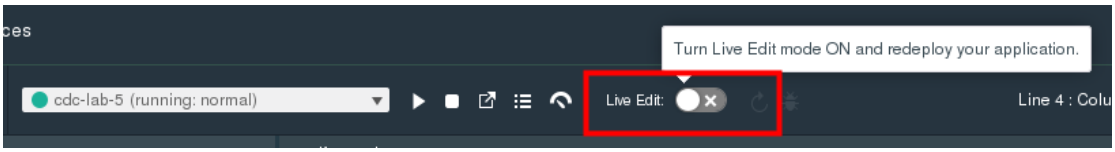
The status bar will change to deploying and after a few minutes, it will change to running.

5. Click the **Popout** button to open the application in a new browser tab.

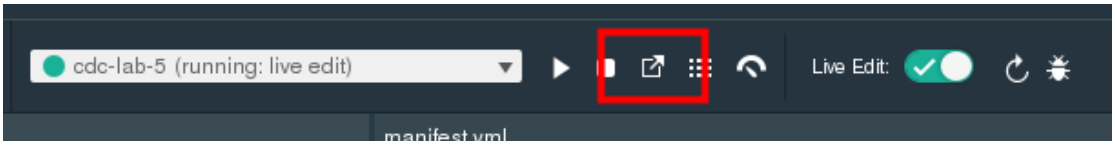


Exercise 5.3.4: Using Live Edit mode

1. Close the **Application** tab and return to the **DevOps Services** tab. Then, click the **Live Edit** toggle switch to enable live editing of the application.



2. Wait for your application to redeploy and click the **Popout** button again to open the application in a new browser tab.

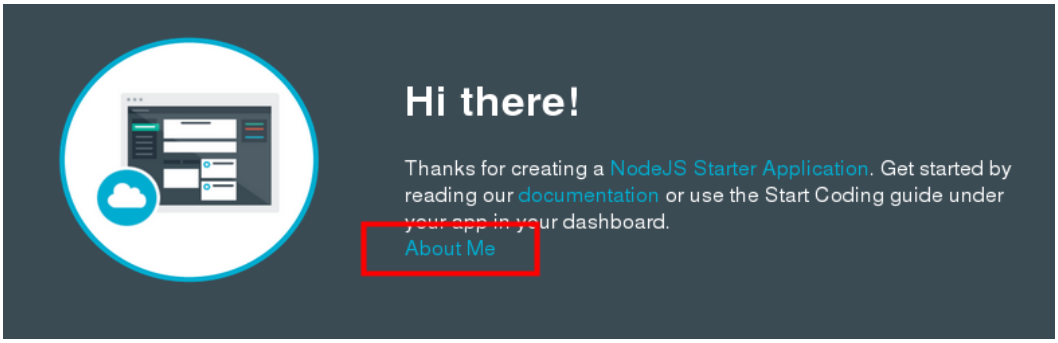


Live Edit allows you to rapidly make changes to your application code and see those code changes without having to redeploy.

3. Leave the **Application** tab open and return to the DevOps Services editor. Click the index.html file and at the end of the <p> tag section, add the following two lines of code:

```
or use the Start Coding guide under your app in your dashboard.
<br />
<a href="about.html">About Me</a>
</table>
```

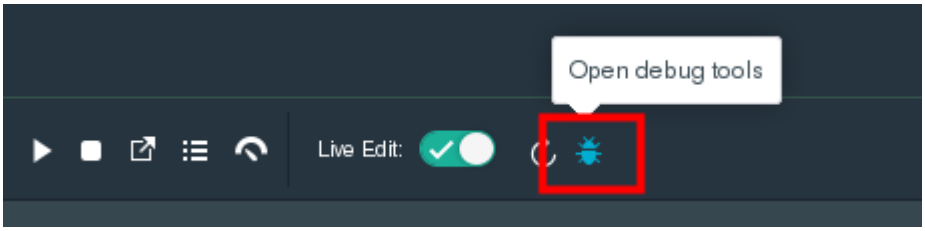
4. Click **Restart** in the Bluemix control pane to restart the node.js server, picking up your recent change to the index.html file. Note that the restart function takes only a second to complete. Switch to the **Application** tab and refresh the page. The code change is now live.



Exercise 5.3.5: Exploring the DevOps Services debug tools

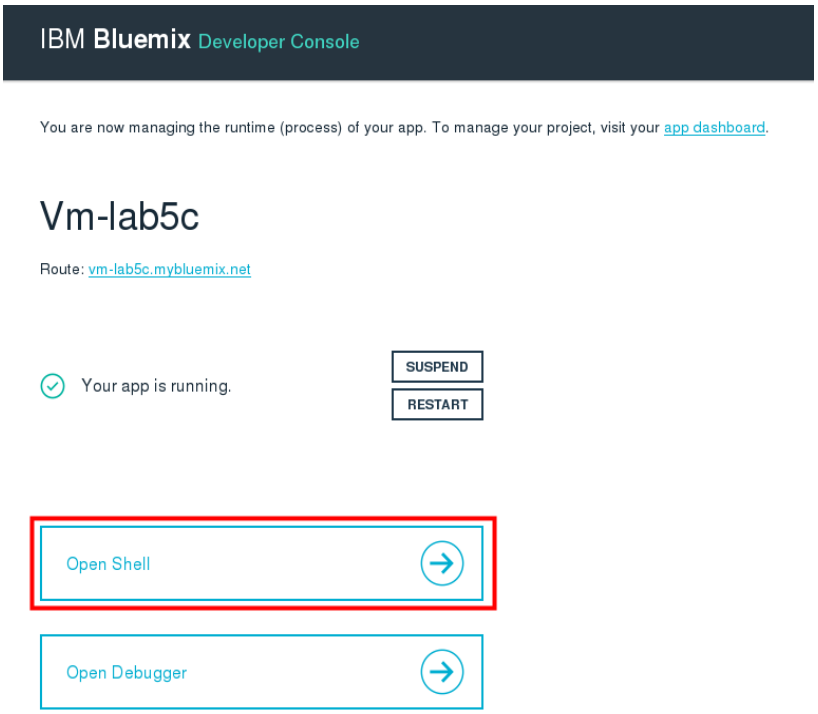
The steps in this exercise require the Chrome browser.

1. Close the **Application** tab and return to the DevOps Services editor. In the Bluemix control pane, open the debug tools. When prompted, enter your IBM id credentials.



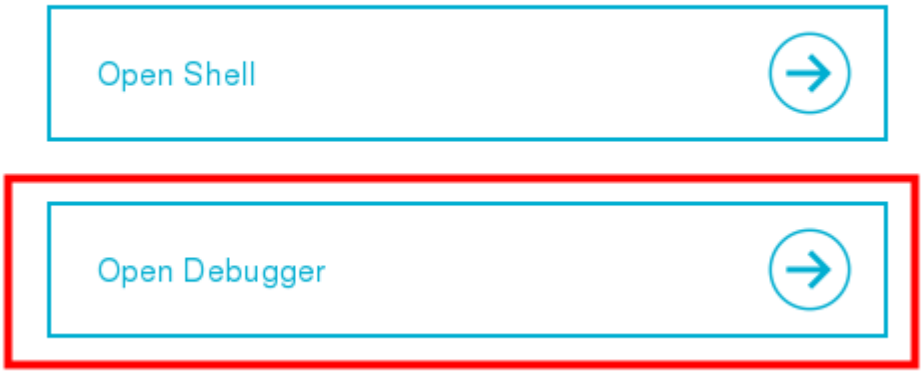
   The IBM Bluemix Developer Console opens and gives you a status of your application.

2. Click **Open Shell**.



   A new tab opens, which gives you a shell prompt on the virtual machine running your application. From here, you can get interesting information about the status of your app and container by using standard Linux commands such as `free` and `top`.

3. Close the tab and return to the IBM Bluemix Developer console. Then, click **Open Debugger**.



A new tab opens and after about a minute, a full debugger appears. Inside the debugger, you can perform normal debug actions such as setting breakpoints, setting watch expressions, viewing the call stack, and stepping through your code during runtime.

Exercise 5.3: Debugging cloud applications