



IBM Cloud Developer Certification Training

Edit and debug Cloud applications using IBM Bluemix
DevOps Services web code editor

Version: 3

Last modification date: 17 February 2016

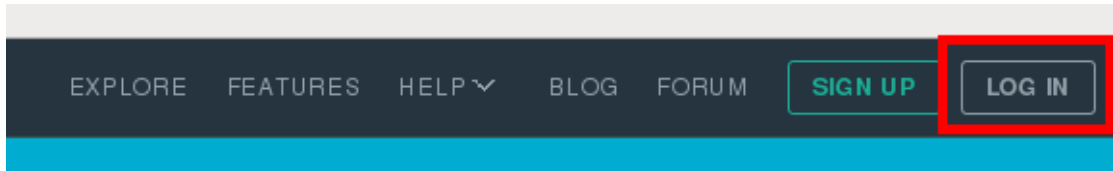
Owner: IBM Ecosystem Development

Exercise 5.3.0 - Lab Prerequisites

- IBM Bluemix account
 - Sign up for Bluemix: <http://bluemix.net>
- IBM DevOps account
 - Sign up for DevOps: <http://hub.jazz.net>
 - Use the same credentials as for Bluemix
- Supported web browsers:
 - Chrome, latest version for your OS
 - Firefox, latest version for your OS or at least ESR 31
 - Internet Explorer, versions 10 and 11
 - Safari, latest version for your OS

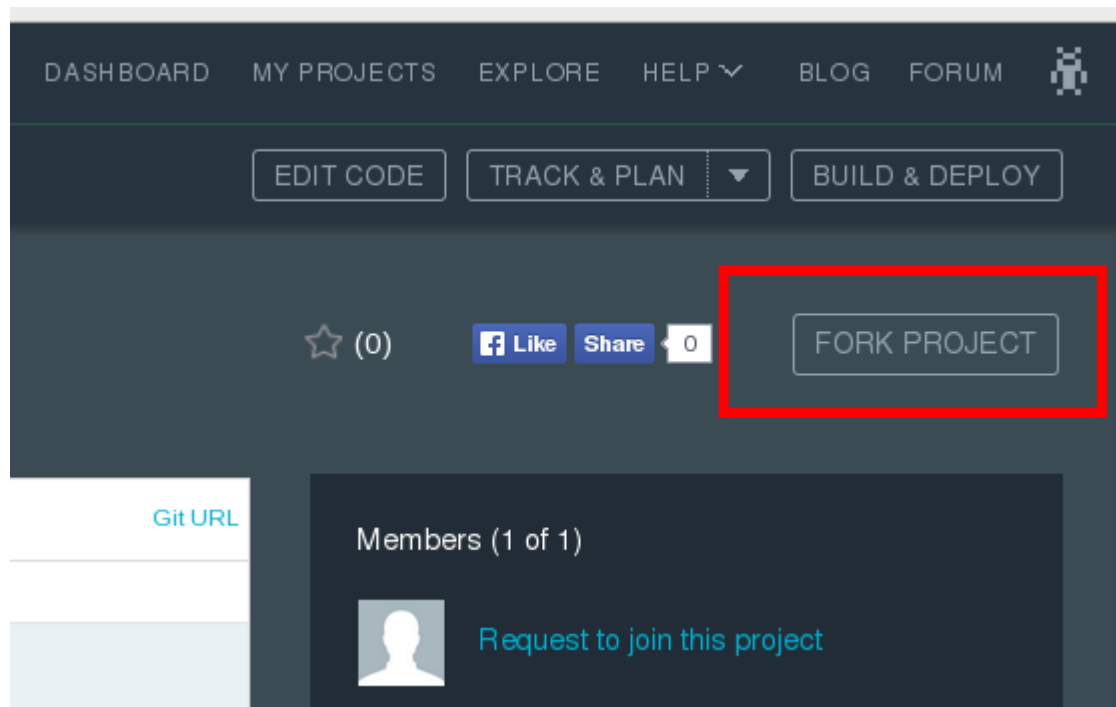
Exercise 5.3.1 - Getting Started

1. Begin by forking the lab project in DevOps Services. Open your browser and navigate to <http://hub.jazz.net>, then click on the Login button in the top right.



Enter your IBM account credentials to complete the logon process.

2. Next, open the URL <https://hub.jazz.net/project/ecosysdevcnc/cdc-lab-5/overview> and click the FORK PROJECT button in the top-right corner.



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

Fork Project

Name your project:

holocron | cdc-lab-5

URL: https://hub.jazz.net/project/holocron/cdc-lab-5

☒ Private project (Invited team members only)

☐ Restrict membership (IBM only)

You can restrict this project's membership because your email address ends with ibm.com. If this project is for **IBM confidential** business, you must select this option and agree to certain conditions. [Learn more](#)

☐ I accept the terms and conditions

☒ Add features for Scrum development

☒ Make this a Bluemix Project

Select a Bluemix space to bill your services to:

Region

IBM Bluemix US South

Organization

vmorris@us.ibm.com

Space

dev

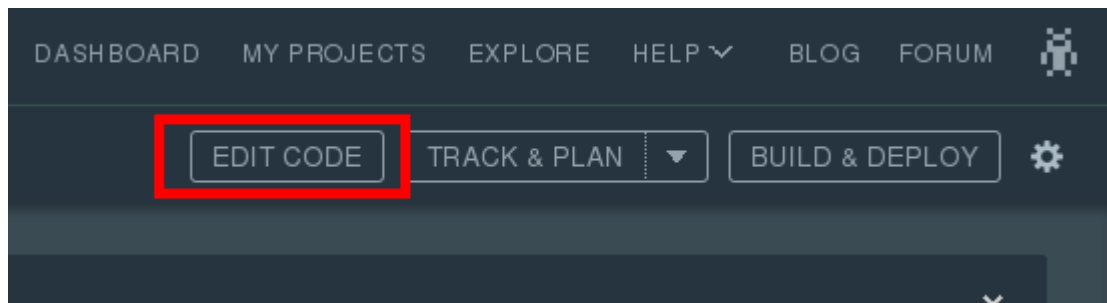
These selections can be changed later in the options for your Project Settings.

CANCEL

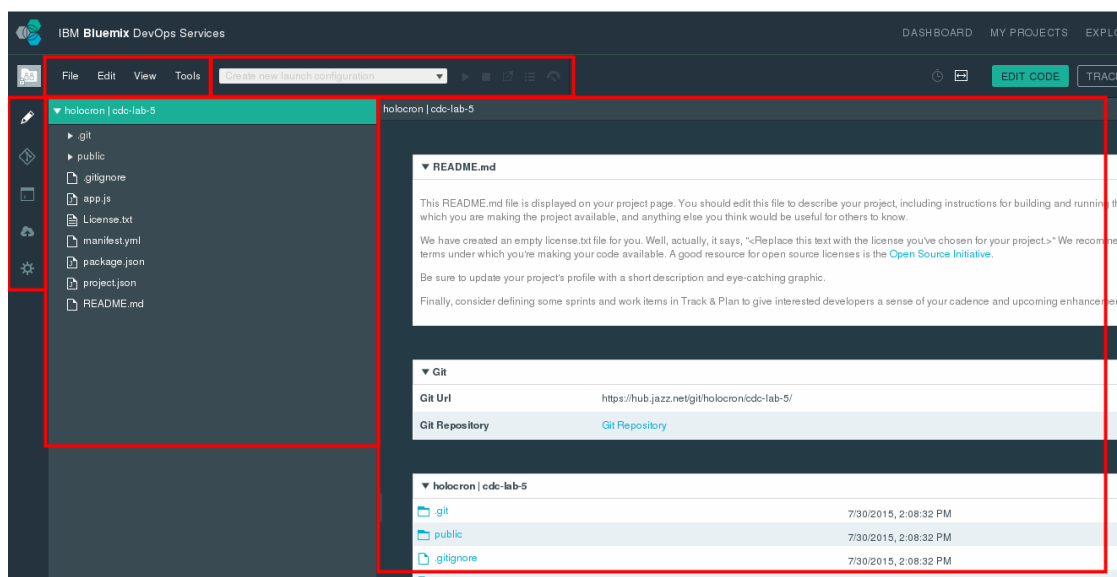
CREATE

Exercise 5.3.2 - Using the DevOps Service Web Code Editor

1. Once you see the message about successfully creating a new project, click on the EDIT CODE button in the top right.



2. 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 filesystem explorer, and to the right, taking up the majority of the screen, is the code editor and preview pane.

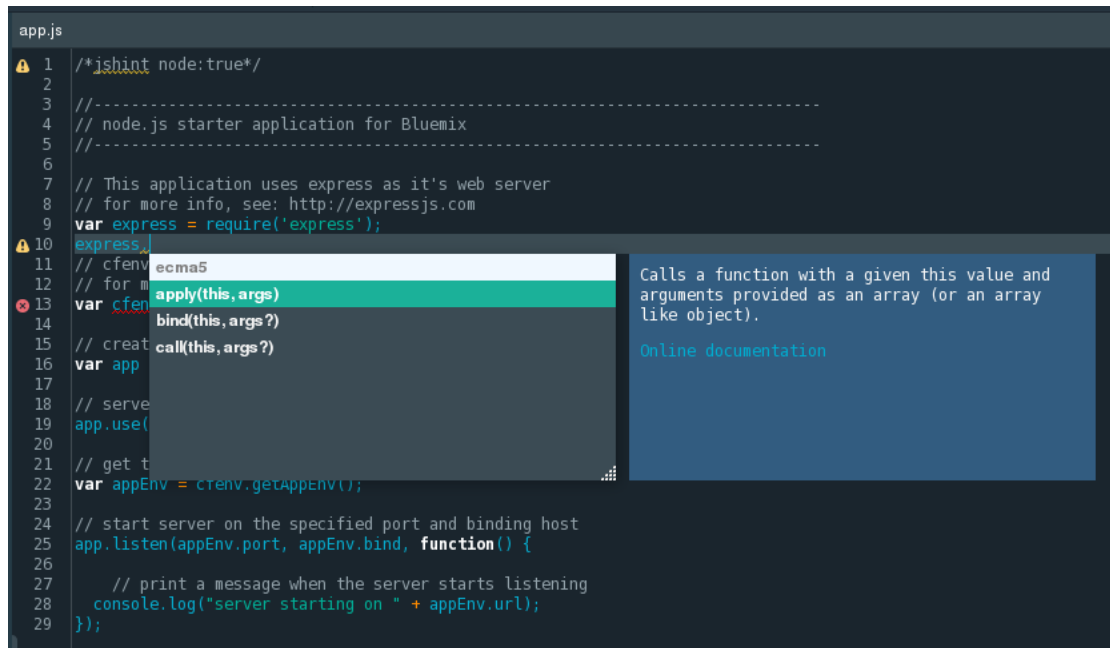


3. Click on app.js to open it in the editor and note how the editor automatically assigns colors based on the inherent knowledge that the file contains JavaScript code.

On line 10, after the `var express` assignment, add the following text (including the period):

`express.`

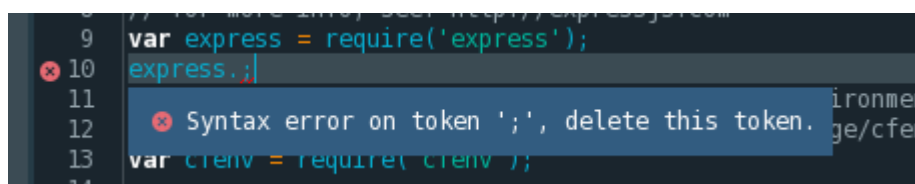
Note how the editor pops up a dialog with suggestions for completing the statement, and also includes text from and links to online documentation for the particular API.



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 rest of this lab.

4. Select the public directory, then click on **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:

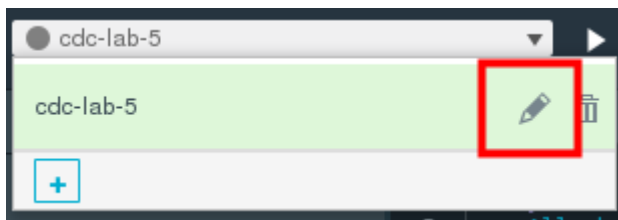
```
<html><body>    All about me!</body></html>
```

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

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

Exercise 5.3.3 - Deploy 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 boxes. These names should match, and combined with the domain will create the Bluemix route through which you can access your running application. You can also change the region for deploying the application with the Target field. This allows you to define multiple Launch Configurations with different names for each region. Keep the current Target and click Save.

Launch Config Name*: cdc-lab-5

Target*: IBM Bluemix US South

Organization*: vmorris@us.ibm.com

Space*: dev

Manifest File: ☒ manifest.yml

Manifest Settings

Application Name*: vm-lab5c

Host: vm-lab5c

Domain*: mybluemix.net

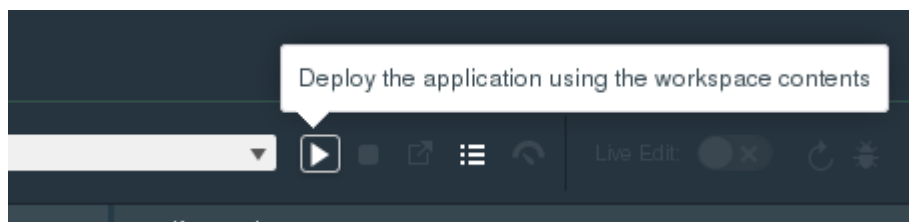
Yellow boxes indicate modified fields, which will override manifest file settings.

Cancel Save Next >

3. These configuration settings override the settings located in a file called manifest.yml. Open this file now in the editor, and overwrite the host: and name: parameters with the same unique name you set in the launch configuration dialog.

```
manifest.yml
1 applications:
2   - disk_quota: 1024M
3     host: vm-lab5c
4     name: vm-lab5c
5     path: .
6     domain: mybluemix.net
7     instances: 1
8     memory: 512M
9
```


4. You are now ready to deploy the application to Bluemix. Click on the Play button in the Bluemix control pane.

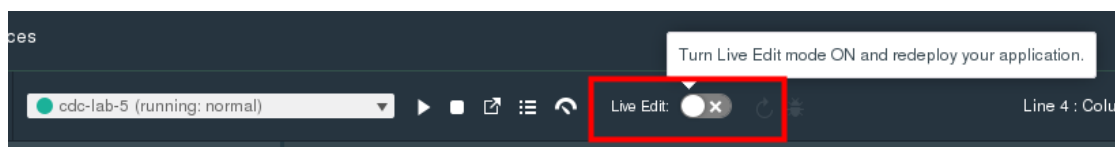


5. The status bar will change to deploying, and after a few minutes will change to running. Click on the popout button to open the application in a new browser tab.

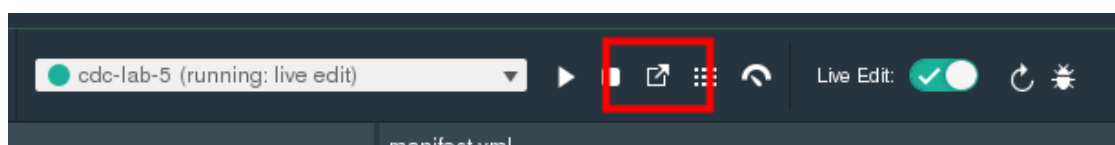


Exercise 5.3.4 - Using the Live Edit mode

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



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



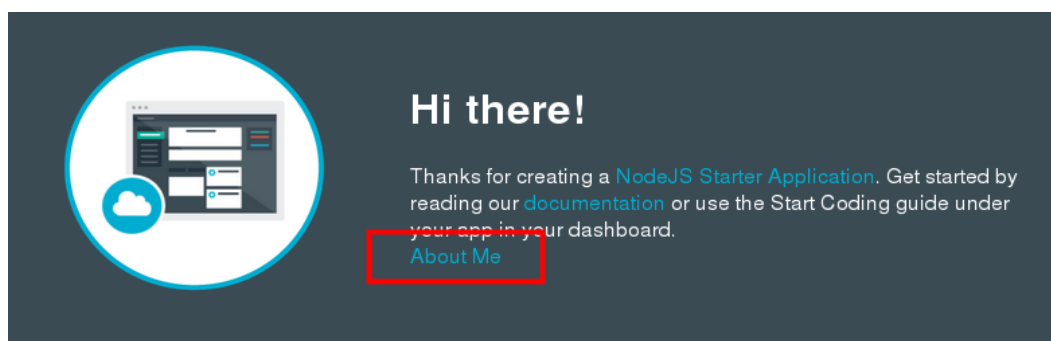
Live Edit allows you to rapidly make changes to your application code, and see those code changes without having to redeploy.

3. Leave the application tab open, and return to the DevOps Services editor. Click on index.html 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

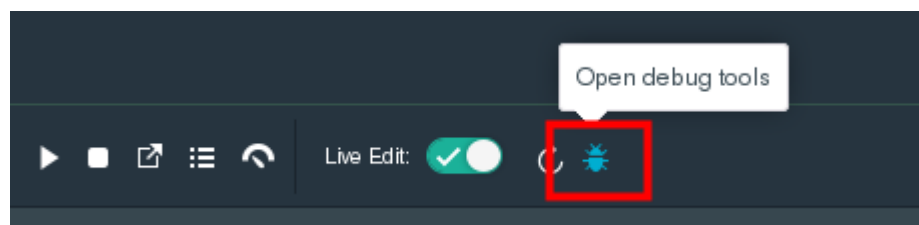
```
dashboard.  
<br />  
<a href="about.html">About Me</a>  
</table>
```

After the change to index.html, switch to the application tab and refresh the page. The content change is immediately live. If you are updating javascript code in the node.js application you need to click the restart button in the Bluemix control pane to restart the node.js server. The restart function takes only a few seconds to complete which is much faster than redeploying the application.

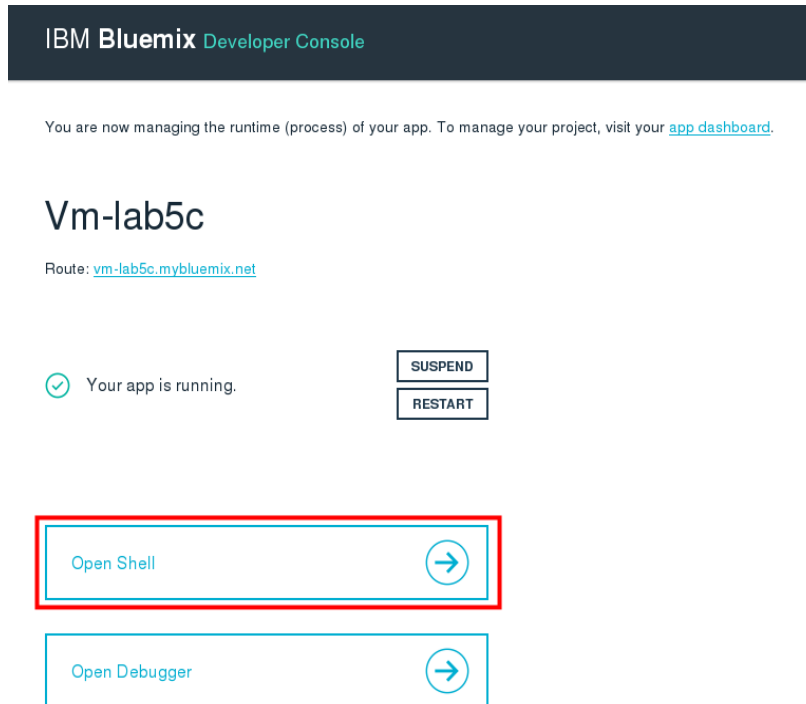


Exercise 5.3.5 - Explore the DevOps Services Debug Tools

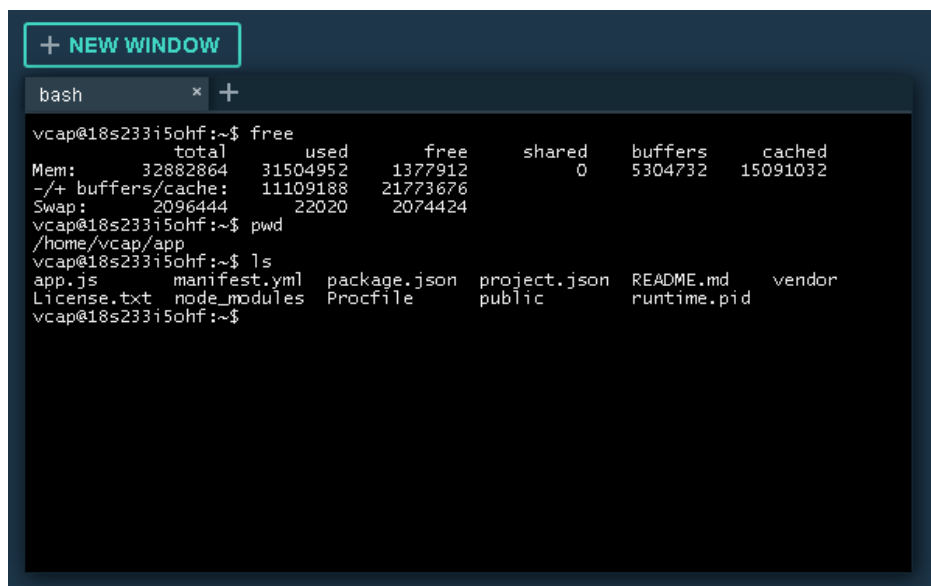
1. (Optional - the remaining steps in this lab require the Chrome browser to complete) Close the application tab and return to the DevOps Services editor. In the Bluemix control pane, click on the button to open the debug tools. When prompted, enter your IBM id credentials.



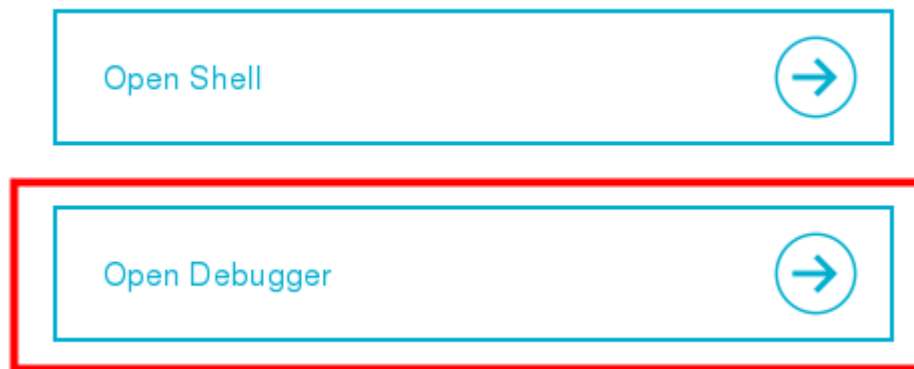
2. The IBM Bluemix Developer Console will open, and give you a status of your application. Click on the Open Shell button.



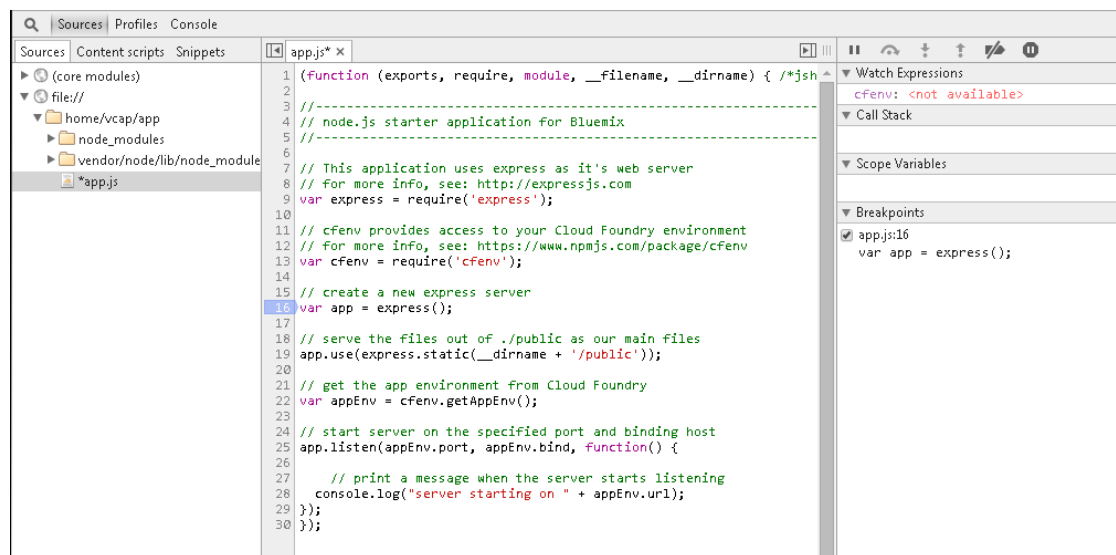
A new tab will open, giving you a shell prompt on the virtual machine running your application. From here, you can get interesting information about the status of your app and container using standard linux commands like `free` and `top`.



3. Close the tab and return to the IBM Bluemix Developer Console, then click on the Open Debugger button.



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



This completes the IBM Cloud Developer Certification Section 5.3 Lab.