Cloud Developers Certification Training

Debugging Applications in Bluemix

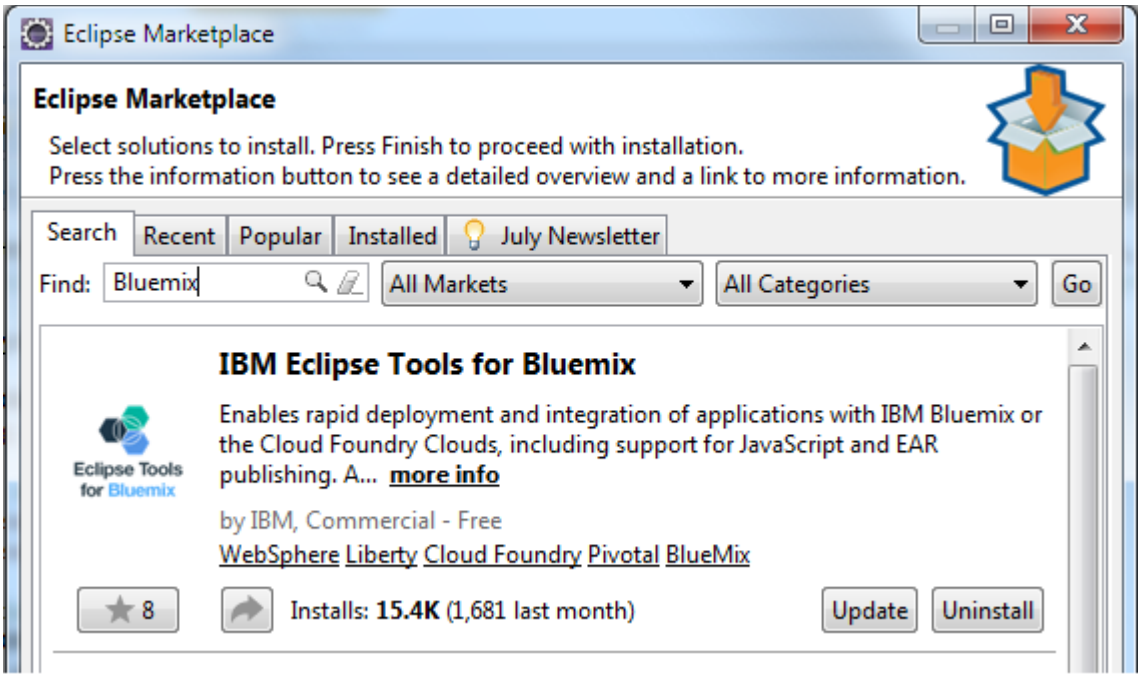
Lab Exercise

Version:	1
Last modification date:	3-Sep-15
Owner:	IBM Ecosystem Development

LAB OBJECTIVE: Debug a cloud ready application in Bluemix. Our objective is to look at the errors encountered while running an application and resolve them. We will debug a Java application and Node.js application.

PRE-REQUISITES :

1. Install Eclipse Luna (4.4) or Kepler (4.3). We expect the audience to be familiar with using Eclipse.
2. Install Bluemix Plugin for Eclipse using Eclipse MarketPlace as shown below :



3. A Java project already deployed to Bluemix.

Debugging Java Application using IBM Java Liberty Runtime

How: Please import SampleWebApp.zip from https://github.com/sandhya9/SampleWebApp.

Benefits: Switch an application instance in Bluemix to Dev Mode to be able to:

- Initiate remote debug sessions with that application instance
- Push incremental file updates to that application instance and see the changes without repushing the app;
- Run and access additional tools inside the app container such as a SSH web console, as supported by each Buildpack's Dev Mode.

Steps:

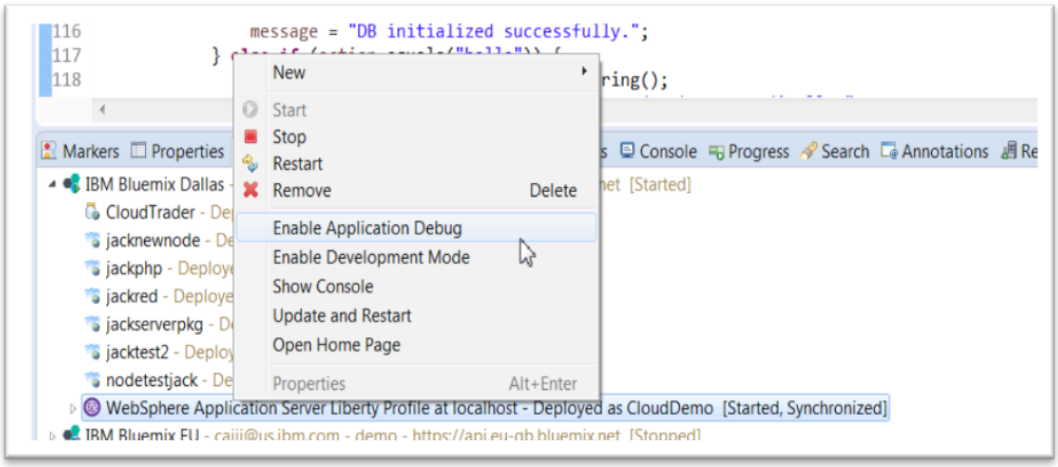
- IBM Bluemix Components need to be installed on Eclipse as described at

  https://www.ng.bluemix.net/docs/#manageapps/eclipsetools/eclipsetools.html

- Next, under the Bluemix server, right-click the application that you want to debug.

- Note: This function cannot be enabled if the application deployment name has an underscore. Change the name before enabling remote debug.

- Click Enable Application Debug. The Progress View shows the status of Establishing debug session for <AppName>. The application will show that it is "Developing, Debugging <AppName>". At this point, the debugger is running and ready to use



- Back to Normal Mode: Switch an application instance in Bluemix from Dev Mode to Normal Mode, so that the application comes back to normal as if nothing happened during Dev Mode.

- You can disable the debug process and leave development mode enabled, if you so choose.

    - Right-click the application.
    - Click Disable Application Debug.
    - A dialog box asks if you want to disable development mode, also. Click Yes or No.

Note: If developer repushes/restages/restarts the application, all the application instances will also come back to Normal Mode. And all file changes made during Dev Mode will be lost.

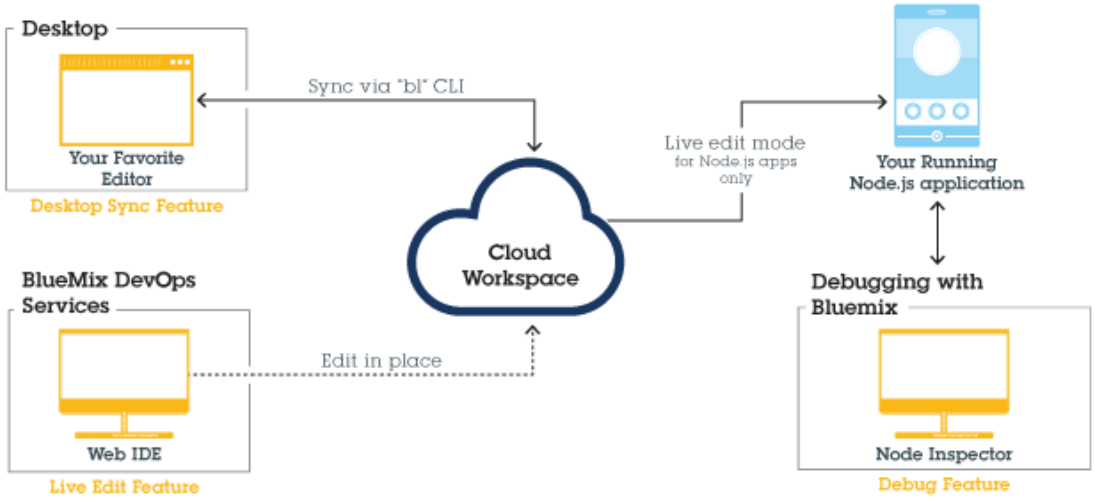
Debugging Node.js using SDK for Node.js

PRE-REQUISITES :

- Sign up for DevOps Services. When you sign up, you'll create an IBM id, create an alias, and register with Bluemix.
- If you don't already have a Node.js project to work with, create one.

- To develop locally from your desktop, download and install the Bluemix Live Sync command-line interface for Windows or for Mac OS X. The Live Sync command-line interface is available only for Windows 7 and 8 and Mac OS X version 10.9 or later.

Steps: Use the Bluemix DevOps service's "Live Edit" feature via Favorite IDE like Eclipse or DevOps Service after turning on dev mode



Benefits: Switch an application instance in Bluemix to Dev Mode in SDK for Node.js to :

- Set breakpoints in the app code to pause execution at a specific line.
- Edit breakpoint conditions to pause execution only when certain criteria are met.
- Inspect the state of local variables and fields.
- View debug output from console.log() calls immediately. This action is faster than monitoring cf logs.
- Use the built-in source code editor to make immediate, yet temporary, changes to the running app code.

Enabling Dev Mode for Debugging in IBM Node JS Build pack

- Allow the buildpack to detect the app start command. The start command must be auto-detected by the buildpack, not set in the manifest.yml file.

- Ensure that the package.json file contains a start script that includes a start command for the app. If the App's manifest.yml file contains a command, set it to null.

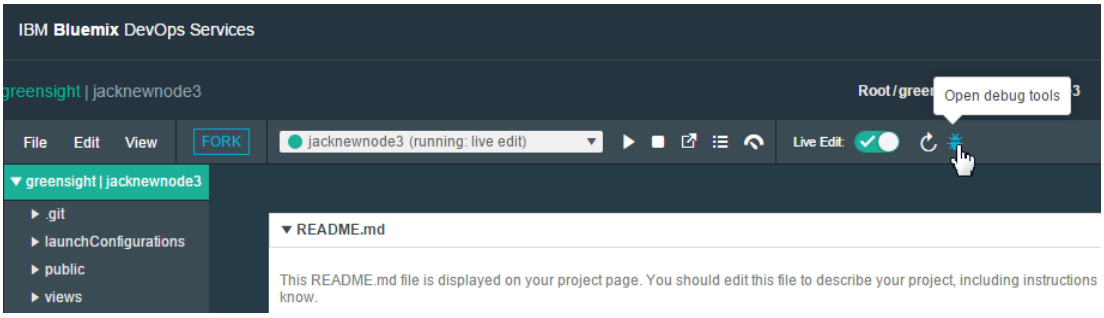
- Set the environment variable. In the App's manifest.yml file, add this variable:
  Env:
  ENABLE_BLUEMIX_DEV_MODE: "true"

- Increase the memory. In the App's manifest.yml file, add 128M or more to the value that is specified for the memory attribute.

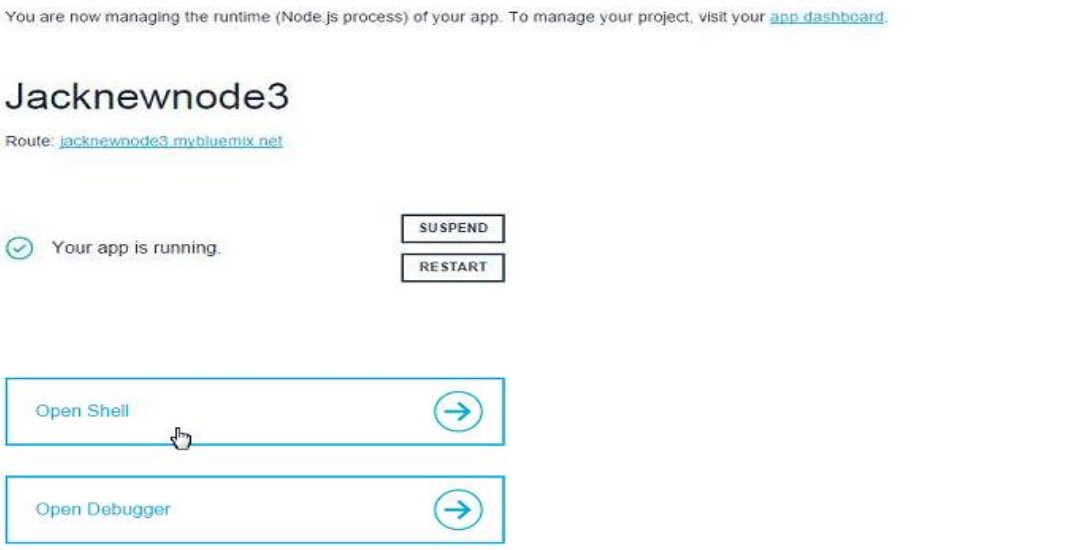
- Install Bluemix Live Debug to use the debug tools.

- Push the app then browse to https://<app-host>.mybluemix.net/bluemix-debug/manage to access the Bluemix debug user interface. When you are prompted, enter your IBM ID and password to authenticate.

- One can also navigate to the debug user interface by clicking on the "Open Debug Tools" button on the DevOps page



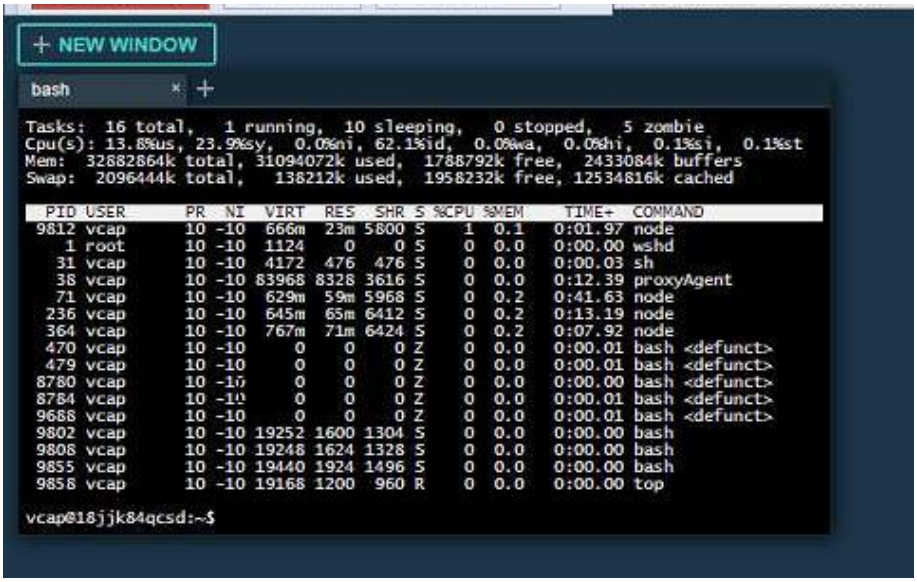
- This is an example of the debug user interface



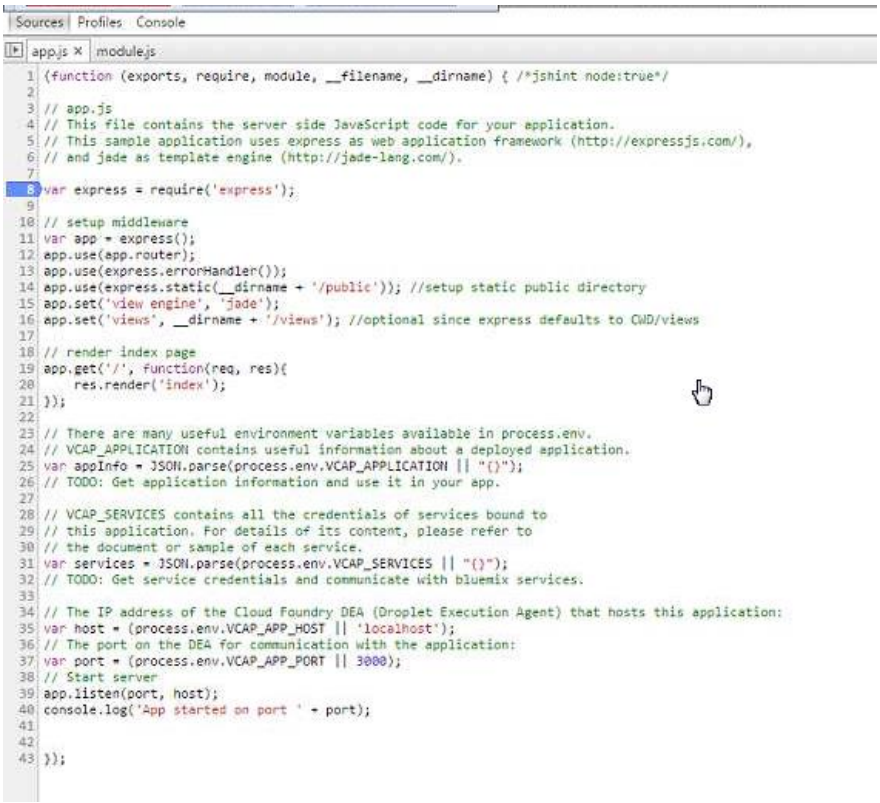
From this DEBUG TOOLS page – we can

1. SUSPEND APP
2. RESTART
3. OPEN SHELL INTERFACE TO THE CONTAINER
4. OPEN DEBUG INSPECTOR (OPEN DEBUGGER)

BELOW IS AN EXAMPLE OF THE CONSOLE WINDOW



BELOW IS AN EXAMPLE OF THE DEBUG INSPECTOR



- Note the Restrictions:
    - Google Chrome is required.
    - The app must use the IBM SDK for Node.js buildpack. Custom buildpacks are not supported.
    - For more info on debuging in SDK for Node.js: https://www.ng.bluemix.net/docs/#manageapps/bluemixlive.html#live_sync-edit

- Back to Normal Mode: Restoring app configurations and disabling Bluemix Live Debug
    - Remove the ENABLE_BLUEMIX_DEV_MODE environment variable from the app manifest.yml file.
    - Restore the app's original start command and memory value.
    - Push the app.

OUTPUTS: Able to Switch IBM Java Liberty Runtime or IBM Node JS Runtime between Dev Mode and Normal Mode to perform Debugging on Cloud Application.