Hands-on Lab

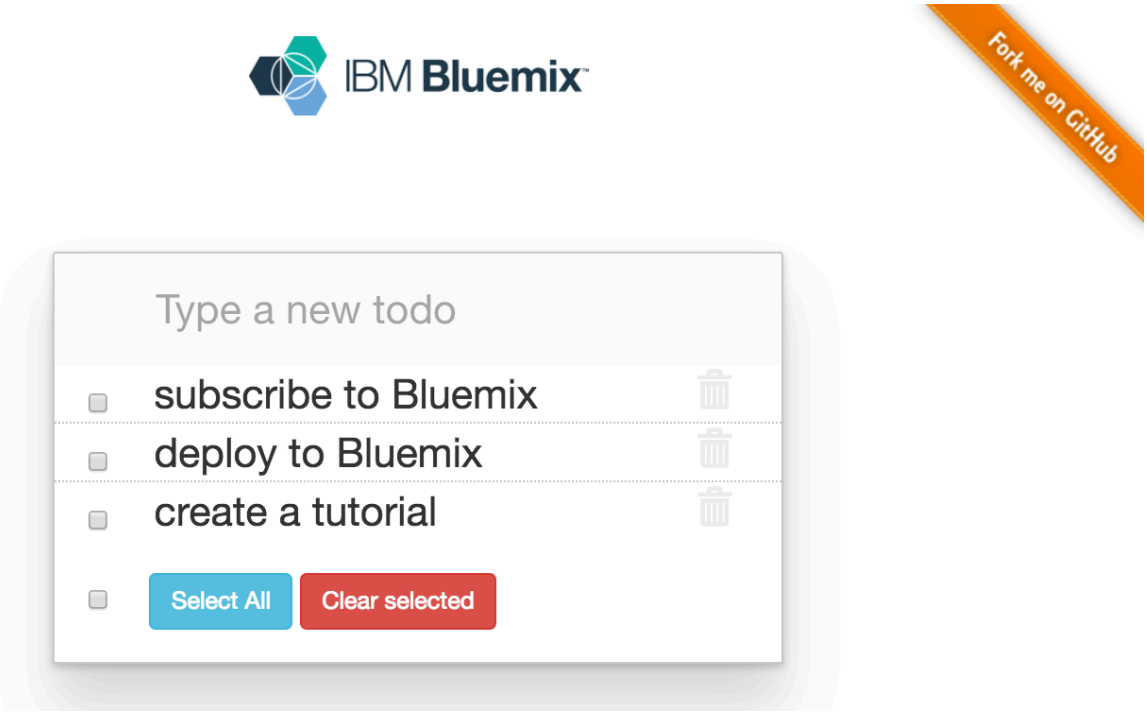
Bluemix

Work with a Node.js app and integrate with Continuous Delivery

Table of Contents

Introduction

In this lab, you'll gain a high-level understanding of the architecture, features, and development concepts related to the Cloud Foundry runtimes and Bluemix services. Throughout the lab, you'll create a sample application built with a CLEAN stack (Cloudant NoSQL database, Express, Angular and Node.js).



Objective

In the following lab, you will learn:

- How to deploy a new Cloud Foundry app based on a Node.js runtime
- How to create a new Cloudant NoSQL DB service to store data
- How to integrate Continuous Delivery with an app

Pre-Requisites

- Get a Bluemix IBM id (https://bluemix.net) or use an existing account

Steps

1. Create a new web application

2. Create and bind a Cloudant NoSQL DB service
3. Enable Continuous Delivery
4. Update the code
5. View the app

Step 1 - Create a new web application

1. Log in to the Bluemix Console (https://console.bluemix.net)
2. Select the Region **US South** to create your application

   Note: This lab is intended to work ONLY in the US South Region where Continuous Delivery Services have been deployed.

3. Go to the **Catalog**
4. In the **Apps** category, select **Cloud Foundry Apps**
5. Select the ***SDK for Node.js*** tile
6. Give your app a unique name and unique host (e.g. todo-[your-initials])
7. Click **Create** to create the application

The SDK for Node.js created a simple "Hello World!" web app that will become our starting point.

Step 2 - Create and bind a Cloudant NoSQL DB service

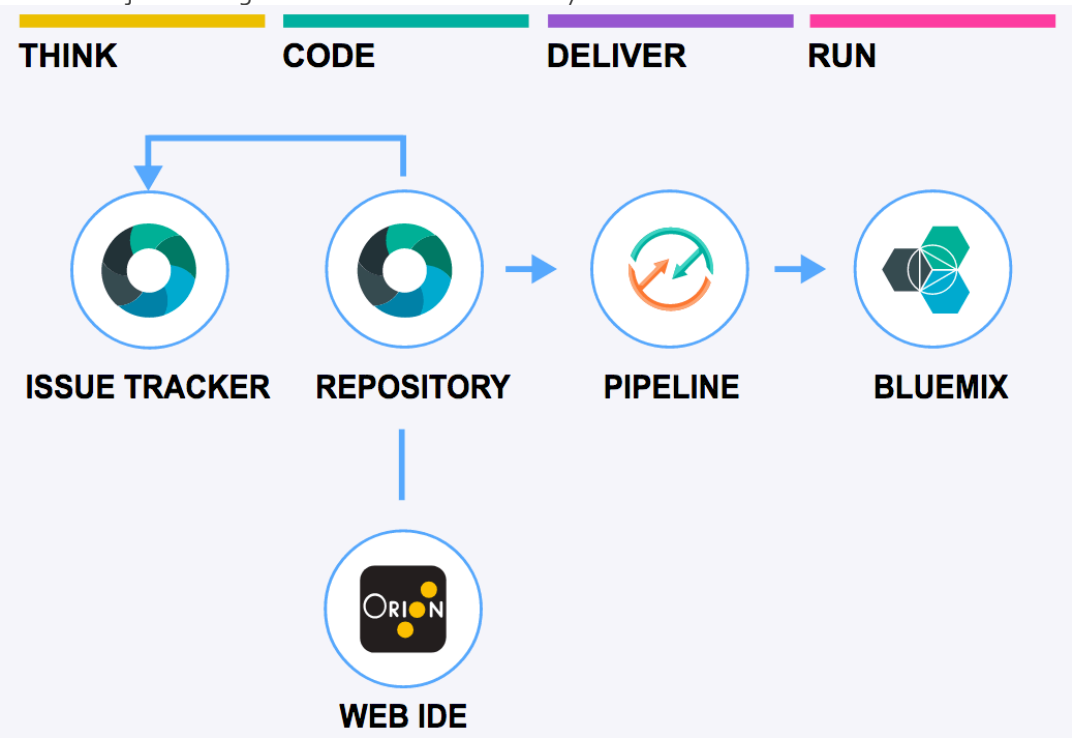
In order to store the todo items we'll need a persistent storage. To do so, we'll use a Cloudant NoSQL database, a JSON document oriented store.

1. Go to the **Overview** section of your app
2. Click **Connect new** to add a service to your application
3. Search for **Cloudant** in the catalog
4. Select the **Cloudant NoSQL DB** service
5. In the details view of the Cloudant service select the app you created from the **Connect to**: drop-down menu to bind this service to the app
6. Give the service a name such as **todo-cloudant-[your-initials]** or you can keep the default
7. Select the free **Lite** plan (default) from the Pricing Plans
8. Click **Create** – Bluemix will provision a Cloudant service
9. Select **Restage** when prompted to do so

   Your application will restart and the service connection information will be made available to your application.

Step 3 - Enable Continuous Delivery

Now let's add a source code repository and an automatic build pipeline to our project. The Git repository and issue tracking is hosted by IBM and built on GitLab Community Edition.



1. In your application **Overview** page, locate the **Continuous delivery** box and click the **Enable** button
2. A new window opens to configure the Toolchain – you can change the default name if desired
3. In **Tool Integrations** at the bottom, select **Git Repos and Issue Tracking**
4. Keep the **Repository type** as Clone and set the **Source repository URL** to https://github.com/optimumsun/todo-app.git
5. Click **Create**, after which the toolchain gets configured and a new Git Repository is created, as well as a Build Pipeline so that your app gets automatically redeployed after every commit

Step 4 – Update the code

1. In the Overview section for the Toolchain click the **Eclipse Orion Web IDE** tile to open up the web editor
2. Open the file tree and click the **manifest.yml** file to edit it
3. Change the name and host values to match the name given for the Node.js app
4. Back in the file tree open the **public/index.html** file and edit line 10 to a title of your choosing
5. On the far left-hand vertical menu click the Git icon to bring up the Git display
6. Enter a commit message in the right-hand panel (i.e. updated manifest and index files)
7. Click **Commit** in the upper right-hand corner of the right pane to commit the altered files
8. Click **Push** in the left-hand panel to push the changes – this automatically triggers the app to rebuild and redeploy
9. Click the back arrow in the left-hand vertical menu to go back to the Toolchain view
10. Click the **Delivery Pipeline** tile to view the Build and Deploy Stages
11. You'll see the Build and Deploy Stages automatically running – wait until the Deploy Stage completes

Step 5 – View the app

1. In the **Overview** section for the app check its status
2. When Running click the Visit App URL link to view the app

Congratulations! You completed this lab. You can get familiar with the application code content.

Source code

Back-end

File	Description
package.json	Lists the node.js dependencies
.cfignore	List of files and directories ignored when calling **cf push**. Typically we ignore everything that can be retrieved with bower or npm. This speeds up the push process.
manifest.yml	Used by Cloud Foundry when pushing the application to define the application environment, connected services, number of instances, etc.
app.js	Web app backend entry point. It initializes the environment and imports the Todo API endpoints
todos.js	Todo API implementation. It declares endpoints for PUT/GET/DELETE (create/retrieve/delete) and handles the *in-memory* storage.

Front-end

File	Description
.bowerrc	Configuration file for the bower web package manager to put our web dependencies under public/vendor
bower.json	Web dependencies (bootstrap, angular)
index.html	Web front-end implementation. It displays the todo list and has a form to submit new todos.
todo.js	Declares the Angular app
todo.service.js	Implements the connection between the front-end and the back-end. It has methods to create/retrieve/delete Todos
todo.controller.js	Controls the main view, loading the current todos and adding/removing todos by delegating to the Todo service

Resources

For additional resources pay close attention to the following:

- GitHub Guides (https://guides.github.com/)
- Get started guides for your favorite runtimes (https://www.ibm.com/blogs/bluemix/2017/03/runtimes-get-started-guides/?social_post=829410659&fst=Learn&linkId=35308736)