

World of Watson

Hands on Labs

Image Analysis

Lab Instructions

1 Building a Simple Image Recognition Application Using Watson Services on Bluemix

This workshop will take you through several of the Watson Services that are available on Bluemix to build a simple image recognition application. Throughout the lab, we will navigate through Bluemix, Github, and the source code of our application in order to demonstrate how apps can be created quickly and easily using the Bluemix platform, and the value of providing Watson and Cognitive capabilities through APIs.

So let's get started. The first thing to do is to build out the shell of our application in Bluemix.

1.1 Creating a shell app in Bluemix

1.1.1 Log into Bluemix

- 1. Go to <https://console.ng.bluemix.net>
- 2. Create a Bluemix account if required,
- 3. Log in with your IBM ID (the ID used to create your Bluemix account)

1.1.2 Watson Developer Cloud GitHub

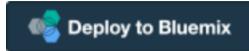
- 1. Go to <https://github.com/aldelucca1/image-analysis>
- 2. Scroll down on the Github landing page and find the “Deploy to Bluemix” button within the ReadMe

Image Analysis

The Image Analysis app let you recognize items in images and speak the resulting description. It uses Visual Recognition to identify objects in images, Machine Translation to translate the description into another language, and Text to Speech to pronounce the resulting translation.

Give it a try! Click the button below to fork into IBM DevOps Services and deploy your own copy of this application on Bluemix.

Demo: <http://image-analysis.mybluemix.net/>



- ___3. From here you will be taken to a Bluemix page, where you will be prompted to name your app. A sample name is provided for you, but feel free to give your application any name you like (if the name is taken by another user you will be prompted to try another name)

Note: use the default settings for Region / Organization / Space on the Bluemix landing page

- ___4. Once you have named your application, click the deploy button to begin the deploy process to Bluemix. During this process, Bluemix will automatically build and deploy our starter application based on the Github repository that we accessed at the start of the lab

APP NAMEtest-image-analysis-lab-app**REGION**

US South

ORGANIZATION

jtarn@us.ibm.com

SPACE

dev

DEPLOY

5. Once the application is finished deploying, you will see a “Success!” message. At this point, scroll to the top of the page and select “Dashboard” from the header bar

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY

Deploy this application to Bluemix

Deploying this app will create a private DevOps Services project for you. [Learn more.](#)



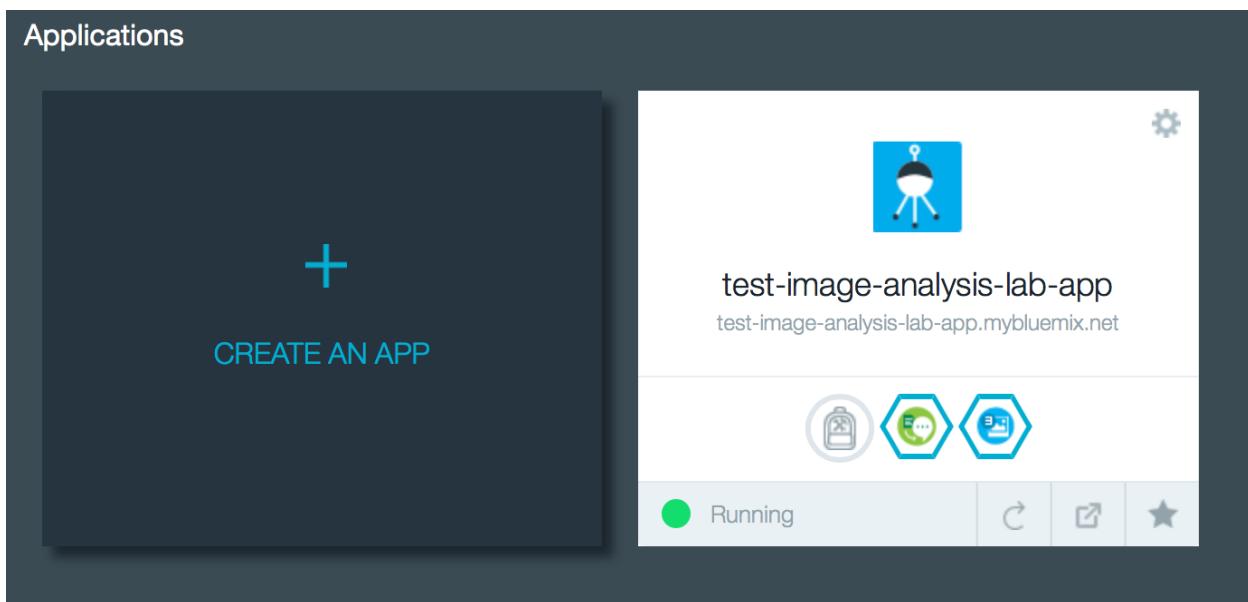
IMAGE-ANALYSIS

GIT URL: <https://github.com/adelucca1/image-analysis>

1.1.3 Add services to the App

1. So far, we have deployed our pre built starter application to Bluemix. We are going to show how easy it is to add additional Watson services to our applications using Bluemix.

On the Bluemix Dashboard, scroll down to find your Image Analysis application within the “Applications” section. From here, click on the application to open the application homepage.



2. Within the application homepage, we are able to see what services we have already included. You will notice that we already have Text to Speech and Visual Recognition built into the application. We are now going to add a third service into the application.

To do this, click the “Add a Service or API” button on the homepage

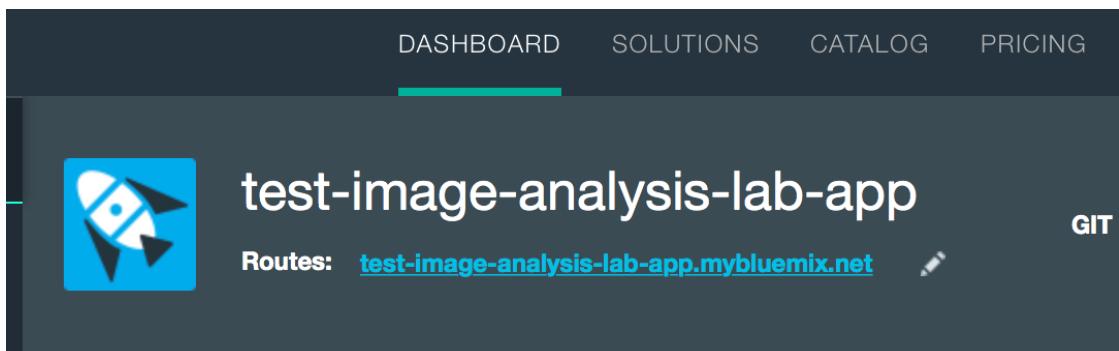
__2. From the list of Watson services, select the Machine Translation service and add it to your application. For the purposes of this lab, all of the default settings of the service will work, so when presented with the Machine Translation details page, select the green “Create” button to proceed.



Note: you may be prompted to restage your application at this point. This is required in order to rebuild the application with the new Machine Translation service that we have added. Select “Restage” to proceed.

1.1.4 Test Out the New Application

__1. Once the services are loaded, you will be able to launch the existing pre-built application. To launch the application, click the link next to “Routes” which should follow the naming convention <your app name>.mybluemix.net



Once launched, you will be able to see the simple image recognition application, that allows a user to select a photo and identify the captured image. When clicking on the speaker button button in the bottom right hand corner you will hear the identified description.

Let's test the application out. Select the  icon in the top right hand corner to prompt for image upload. On the desktop, we have provided some sample images in order to test out the application.



We are going to demonstrate how easy it is to use the Watson services on Bluemix to add functionality to existing applications. Our current application can identify images and read out that identification using audio. However let's say that we wanted to be able to identify these images for a wider user base, which requires translation into other languages.

Luckily, we've already started the process to do this.

To fully implement the ability to translate these descriptions in our application, we are going to edit our application code to add the Machine Translation service that we loaded in earlier.

1.2.5 Modify the existing application

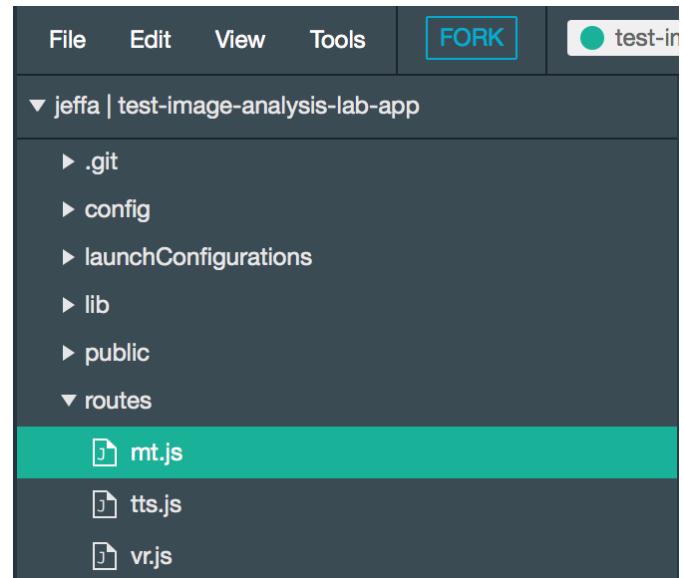
1. Let's edit our source code. Back on the application home page in Bluemix, you will see a link to the Github repository, and a button to 'Edit Code'

click on Edit Code



2. Clicking on Edit Code will take you to the Jazz Hub repository, which will allow us to edit and push new versions of our code to the application.

Within the Github repository, navigate to routes folder and select **File -> New -> File** and name the new file **mt.js**



3. Open up **mt.js** and paste in the following

```
"use strict";

var fs = require("fs"),
    extend = require("util")._extend,
    watson = require("watson-developer-cloud"),
    bluemix = require("../config/bluemix");

module.exports = function() {

    var machineTranslation =
watson.machine_translation(extend({
        version: "v1",
        username: "<<service_username>>",
        password: "<<service_password>>",
    }, bluemix.getServiceCreds("machine_translation")));

    return {
        translate: function(req, res) {

            var params = {
                text: req.body.text,
                to: req.body.to || "es",
                from: "en"
            }
        }
    }
}
```

```

    };

    machineTranslation.translate(params, function
(error, result) {

    if (error) {

        return res.status(error.error ?
error.error.code || 500 : 500).json({ error: error });

    } else {

        return res.json(result);

    }

}) ;

}

} () ;

```

4. Open up your **app.js** and import the newly created **routes/mt.js**. This can be done by adding the following to line 22:

```
var mt = require("./routes/mt");
```



The screenshot shows a code editor with a sidebar containing three files: .gitignore, app.js, and License.txt. The app.js file is currently selected and highlighted in green. The code in app.js starts with several lines of imports (express, vr, tts) and then includes the line: `var mt = require("./routes/mt");`

```

18
19 var express = require("express"),
20 vr = require("./routes/vr"),
21 tts = require("./routes/tts");
22 var mt = require("./routes/mt");
23

```

5. Finally, configure the route in your **app.js** by adding the following to line 31:

```
app.post("/translate", mt.translate);
```

```
28 app.post("/recognize", vr.recognize);
29 app.get("/voices", tts.voices);
30 app.post("/speak", tts.speak);
31 app.post("/translate", mt.translate);
```

1.2.6 Deploy

The last step in order to complete our application is to deploy our changes to Bluemix. To do this, we need to push our new code to the application. In the code editor screen, switch to the Git view, the 2nd icon in the left navigation bar.



Locate your change to app.js file and the routes/mt.js file. Select the check box on both files, add a commit message, and click “Commit”.

The screenshot shows the Bluemix Git interface. At the top, there's a header bar with a search field containing "Working Directory Changes" and several icons for file operations like copy, paste, and delete. Below the header is a commit message input field with the placeholder "Add machine translation". Underneath the message field are two checkboxes: "Amend previous commit" and "Select All". The "Select All" checkbox is checked, and the status "0 files selected" is displayed. A list of files follows, each with a selection checkbox and a detailed view icon (a plus sign inside a circle). The files listed are: ".cignore", "app.js", "launchConfigurations/watson-recognizer-aadelucc-1318.launch", "project.json", and "routes/mt.js". On the far right of the interface, there are additional icons for file operations.

Finally, click “Push” to send your changes from this workspace to the main repository.

Active Branch (master) Sync

Working Directory Changes
Nothing to commit.

Outgoing (0) Push ▾

No Changes

Incoming (0) Fetch ▾

No Changes

History

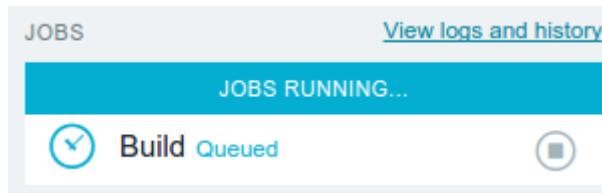
Add machine translation
AI DeLucca on April 29, 2015 at 1:25:52 PM EDT
[more ...](#)

Note: While this may show as overly complicated, we show it here to illustrate you can have exactly the same source management practices you could have your local environment connected to a Git repository, in the Bluemix DevOps environment. In fact, there is even an easier way to deploy changes! (Not pictured here – ask your assistants in the lab!)

- 1. Now that we have added the functions needed to call our Personality Insights, it is time to update the application. Click “Build and Deploy” button in the top right.



- 2. On the “Build and Deploy” page, click the “Play” (>) button which will kick off the deploy to Bluemix



And wait for the jobs to finish:



1.3 Test

- 1. Test the Final Application

To test out our application, navigate back to your application homepage on Bluemix. Select the URL next to “Route” in the same way that we launched our previously unfinished application before.

The new application will perform the same functions as our previous version, but this time you will see translation for the images as well.

Congratulations, you have completed the Image Analysis Lab! We encourage you rate your experience with the labs in a 1 minute survey that can be accessed here: <http://www.surveygizmo.com/s3/2131113/World-of-Watson-Lab-Feedback-Form>