Watson Lab Setup & Intro to Bluemix

Overview

The workstations, used for this lab, need to be updated with the latest code and documentation. This document tells how to do this update in preparation to begin working on the lab exercises.

As part of a brief introduction to Bluemix ecosystem we will also highlight some of the Watson services available today.

Resources

Lab code is on GitHub
https://github.com/WatsonDevCloud/HelloWorld
https://github.com/WatsonDevCloud/Visualization
https://github.com/WatsonDevCloud/NewsAnalyst

Bluemix web console
https://console.ng.bluemix.net/

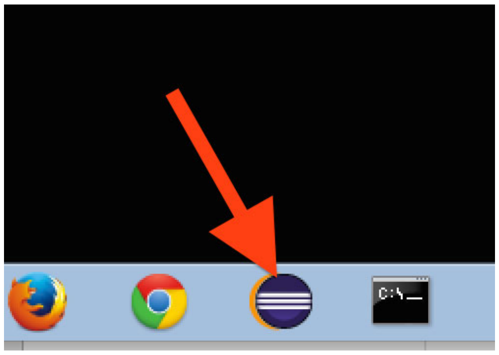
Lab Setup

Windows login

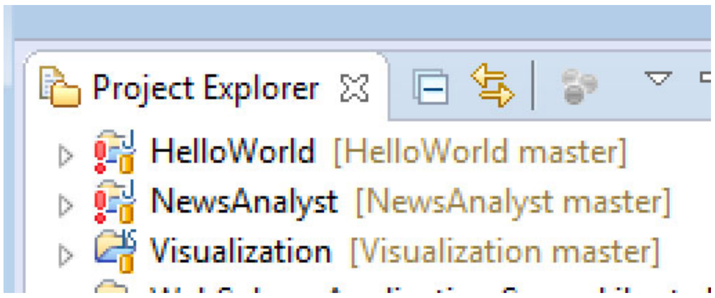
The username is ***IBM*** and the password is ***watson1234***.

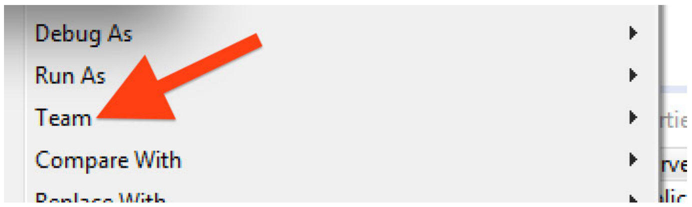
Update Eclipse Projects

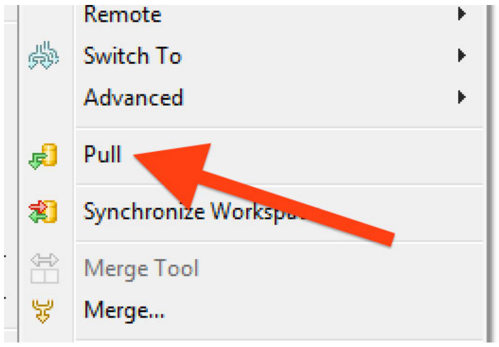
We will be using Eclipse as the IDE for the labs. To start Eclipse, click on the icon in the task bar.



After Eclipse starts, Right-click on the ***HelloWorld*** project and select ***Team>Pull***. This will pull the project from GitHub. The full lab is contained at the GitHub URL listed in the resources section of this document.


Project Explorer
HelloWorld [HelloWorld master]
NewsAnalyst [NewsAnalyst master]
Visualization [Visualization master]





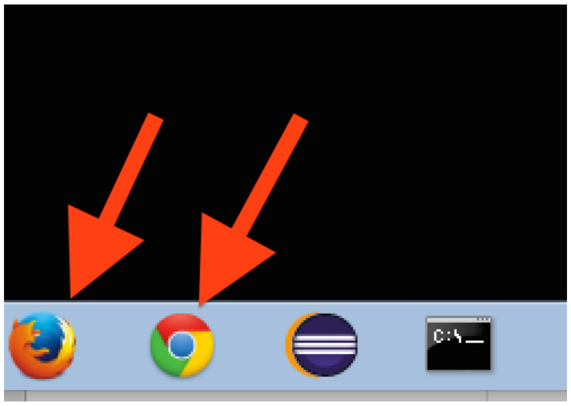
Right-click on the ***NewsAnalyst*** project and select ***Team>Pull***.

Right-click on the ***Visualization*** project and select ***Team>Pull***.

The lab projects are now up to date.

Start Browser

Both the Chrome and Firefox browsers are installed. Open your preferred browser using the icon in the task bar. We will be using a browser to test our apps and to access the Bluemix web console.

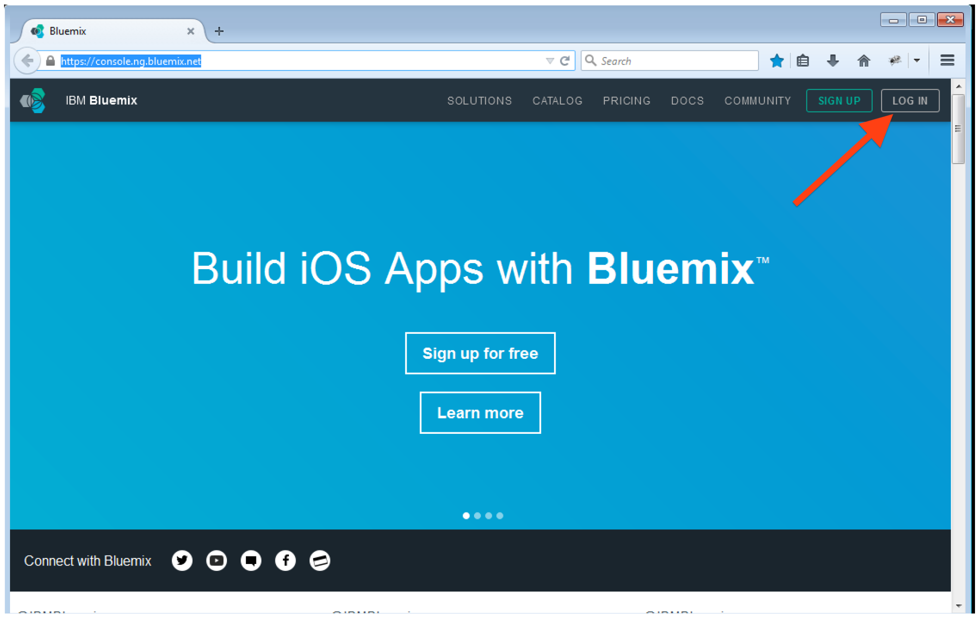


Intro to Bluemix and Watson Services

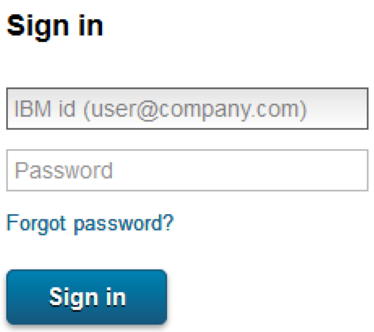
Bluemix Default App

Your browser should open to the Bluemix home page. If not, use this link

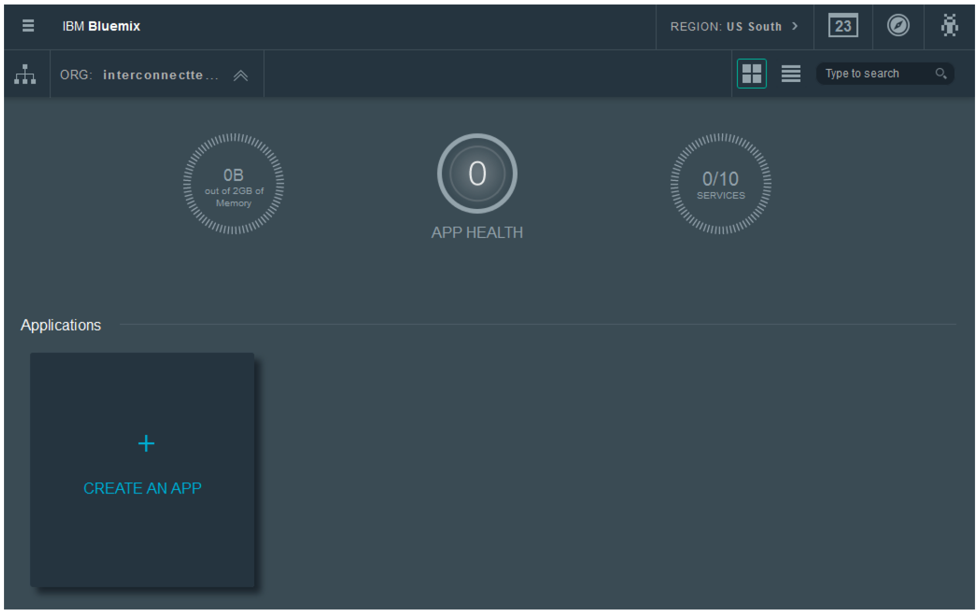
Click on the ***Log In*** button and use the provided username and password to authenticate. If you don't see the *Log In* button, expand the width of the browser window until you do.



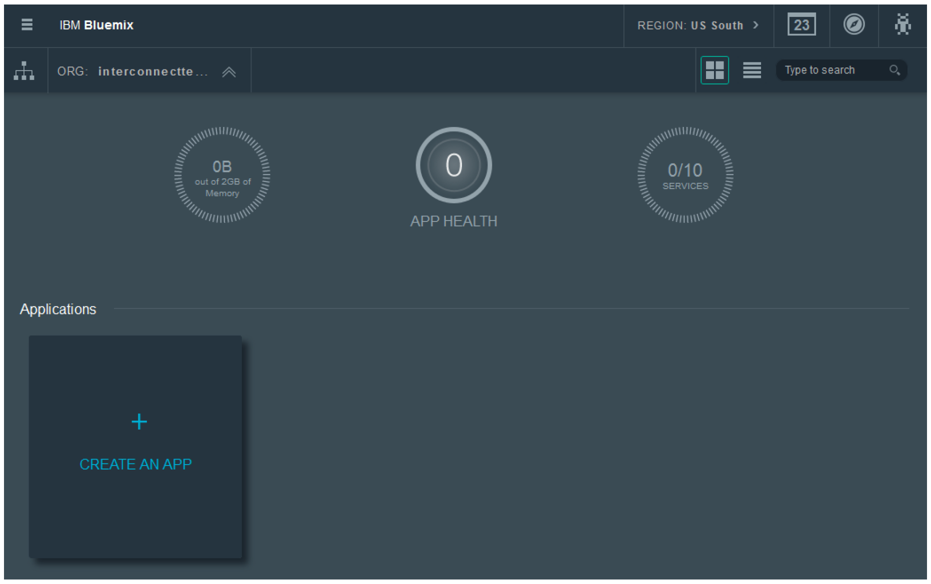
Enter the provided credentials in the *Sign In* page.



After authentication you will end up on the *Dashboard* page. The *Dashboard* provides a high level overview of the applications we have deployed and the services they are using. Since we have no apps, at this point, we are not using any memory or services.

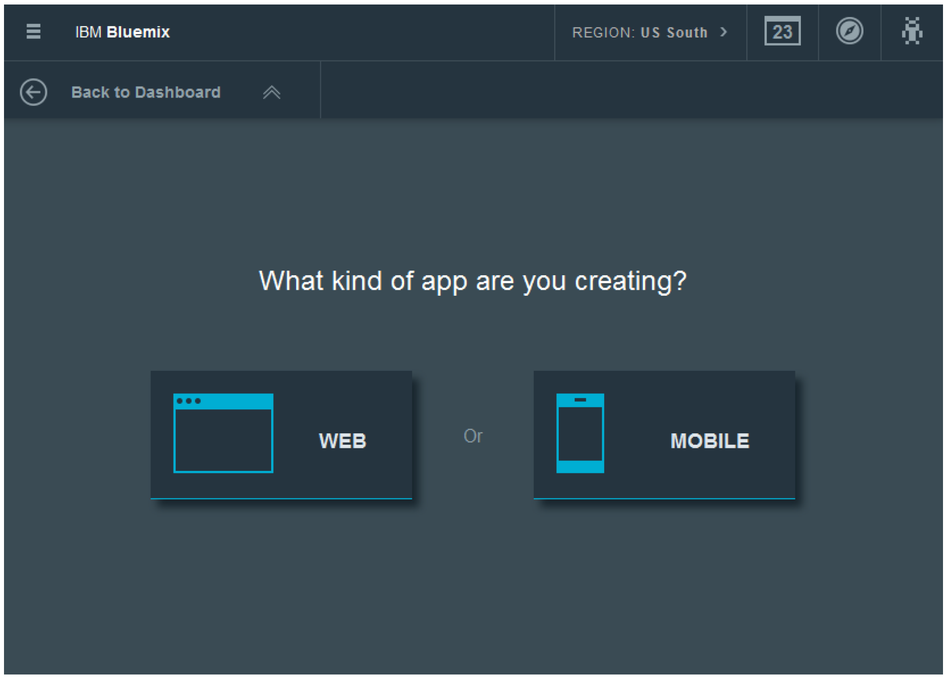


Let's create an app now. Click on ***Create An App*** in the *Applications* section of the *Dashboard*.



For this exercise we are going to create a web app. Click on ***Web***.

For those curious, the *Mobile* apps provide a way to quickly get started with using mobile services for shared data, push, and server-side scripting. Includes SDKs for Android, iOS, and JavaScript. The scope of mobile apps is too large for this lab session.



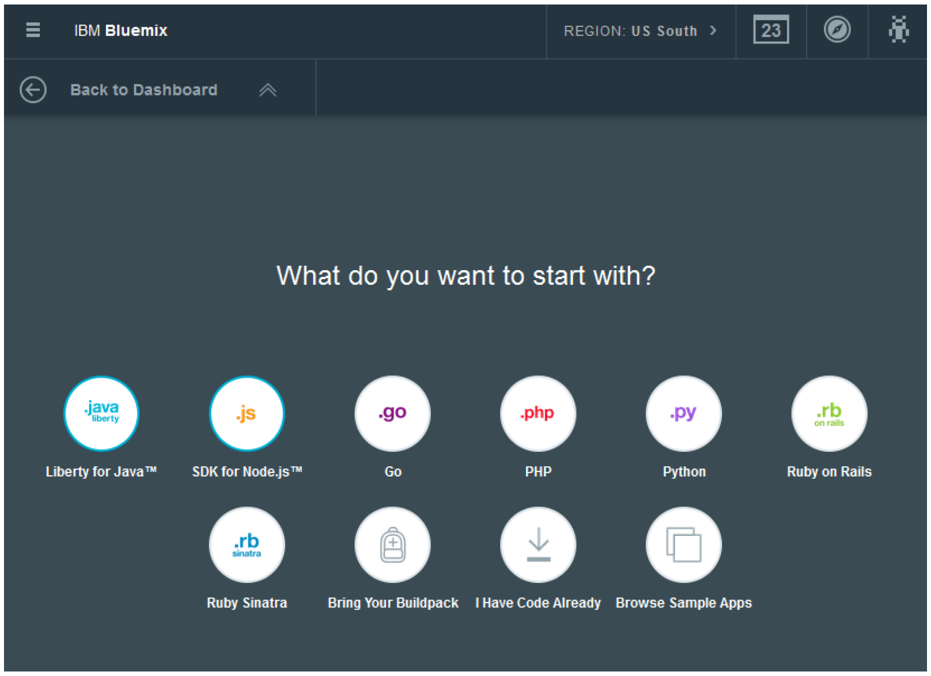
Now you are presented with a list of *Buildpacks* to select from. *Buildpacks* provide framework and runtime support for your applications. *Buildpacks* typically examine user-provided artifacts to determine what dependencies to download and how to configure applications to communicate with bound services. Essentially a Buildpack provides the container in which to run your app. For example, *Liberty for Java* provides a JEE container in which to deploy your war or ear files.

If you don't see a *Buildpack* you like, *Bring Your Buildpack* provides documentation on how to use one of the community provided *Buildpacks* that are available on GitHub.

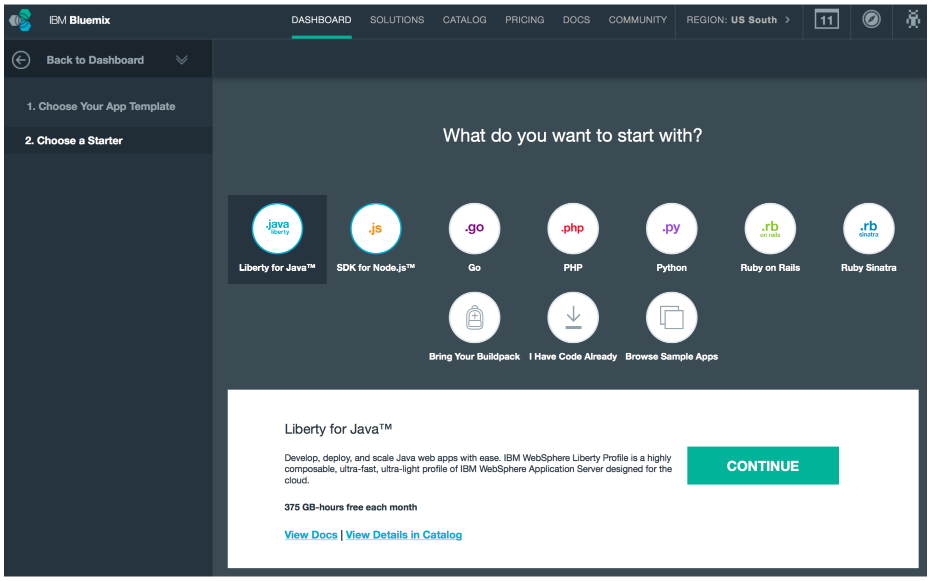
I Have Code Already provides documentation on how to install your app without creating the app in the Bluemix web console. We will be doing this in a later lab.

Browse Sample Apps provides you with a list of *Boilerplates*. A *Boilerplate* is a sample app, built on one of the standard *Buildpacks*, which comes bundled with some Bluemix services.

Click on the ***Liberty for Java*** *Buildpack*.



Click on ***Continue***.



Liberty for Java™

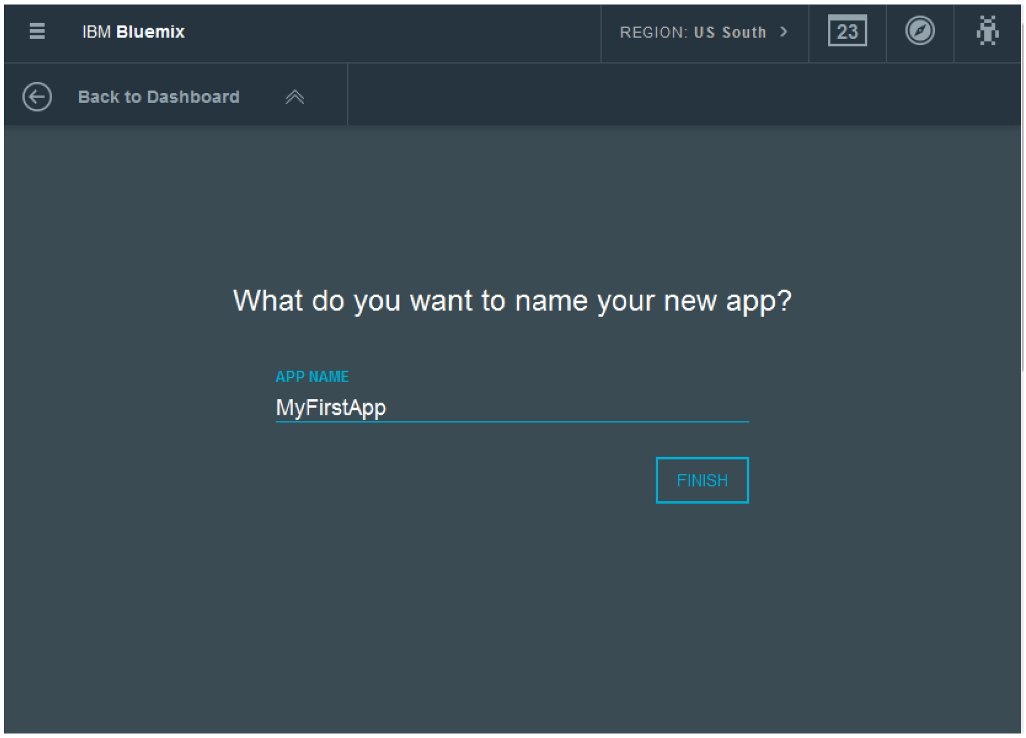
Develop, deploy, and scale Java web apps with ease. IBM WebSphere Liberty Profile is a highly composable, ultra-fast, ultra-light profile of IBM WebSphere Application Server designed for the cloud.

375 GB-hours free each month

Now we have to name our app. When using the web interface to create an app, the app name becomes the host name for the app. For example, an app name of

“MyFirstApp88" will be hosted at http://myfirstapp88.mybluemix.net. The host name needs to be unique in the mybluemix.net domain.

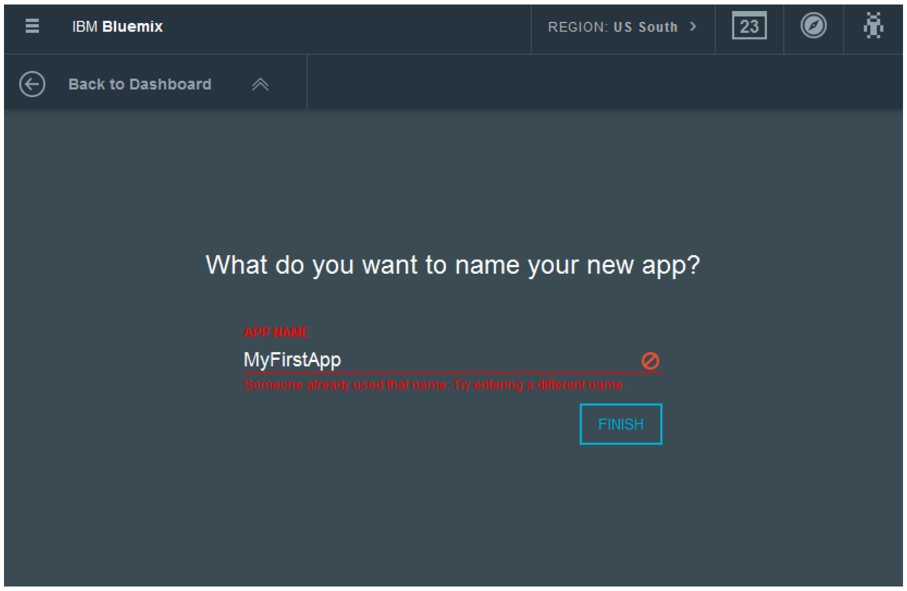
Enter a name for your app, in the *App Name* field, and click ***Finish***. Try to avoid spaces and apostrophes in name to make your life easier.



What do you want to name your new app?

APP NAME
MyFirstApp

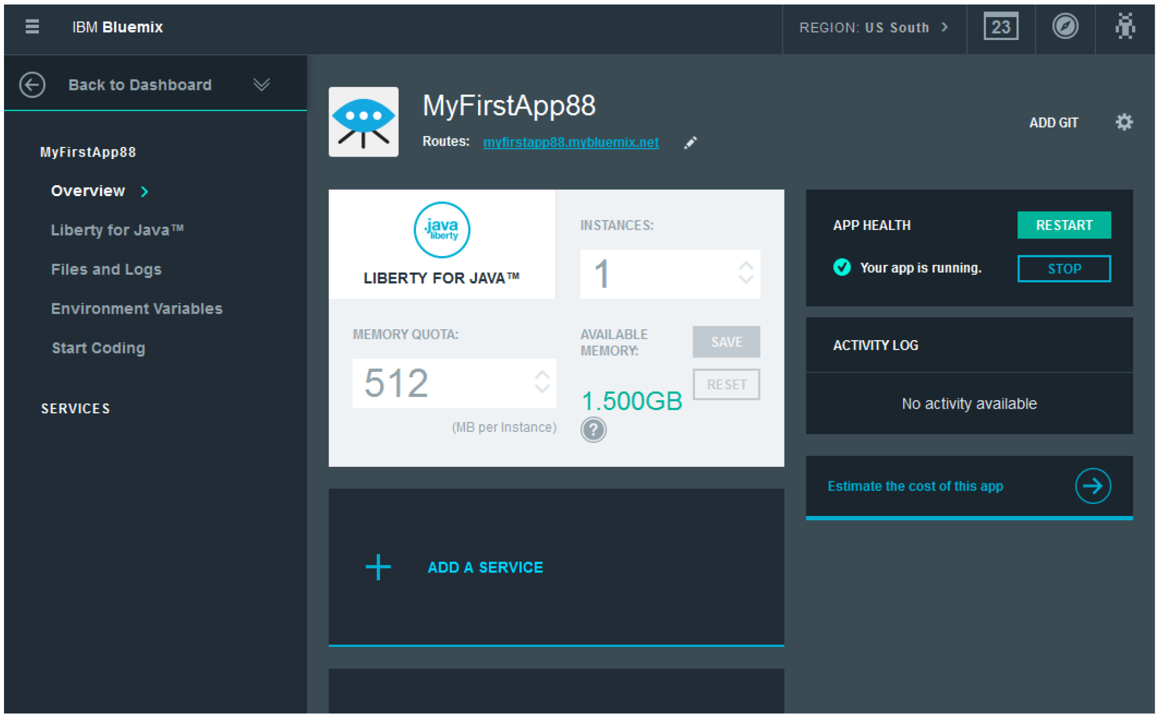
If your app name wasn’t unique then you will see the error page. If you see this, then modify the name and click finish until it accepts the name.



What do you want to name your new app?

APP NAME
MyFirstApp
Someone already used that name. Try entering a different name.

Once Bluemix accepts our app name, a skeleton Liberty for Java app is automatically deployed and the app overview page is displayed. This page contains a lot of information that we will go over now.



MyFirstApp88
Routes: myfirstapp88.mybluemix.net

INSTANCES: 1

MEMORY QUOTA: 512 (MB per Instance)

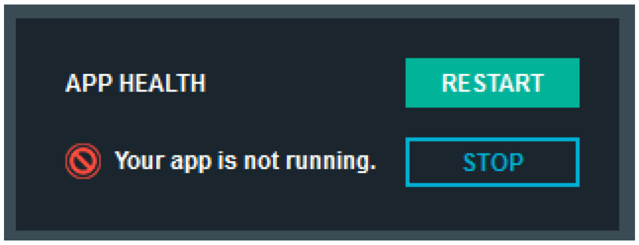
AVAILABLE MEMORY: 1.500GB

Your app is running.

No activity available

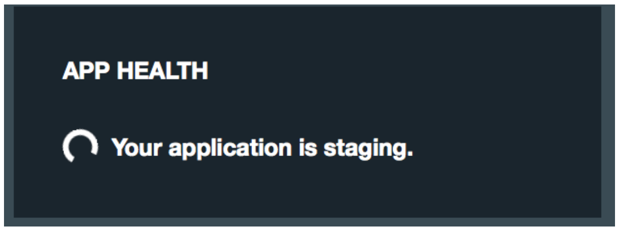
The App Health box tells you the current state of your app. Since we just deployed the app it probably tells us that it is not running.

Note: If you manually stop your app, using the *Stop* button, the state changes to *Stopped*. A Stopped app releases its memory resources. This can be helpful if you are trying to deploy another app and are out of resources.



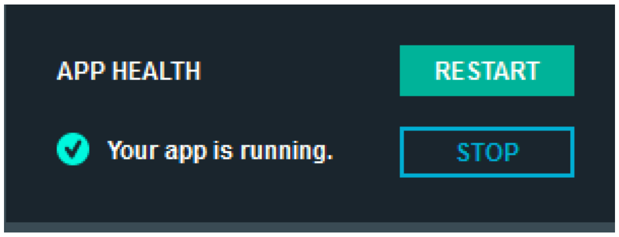
Your app is not running.

During the app deployment process it may also tell you it is staging.



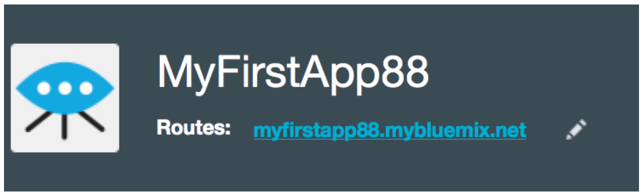
Your application is staging.

Once it is fully deployed, the state changes to running. We now have an operational app.

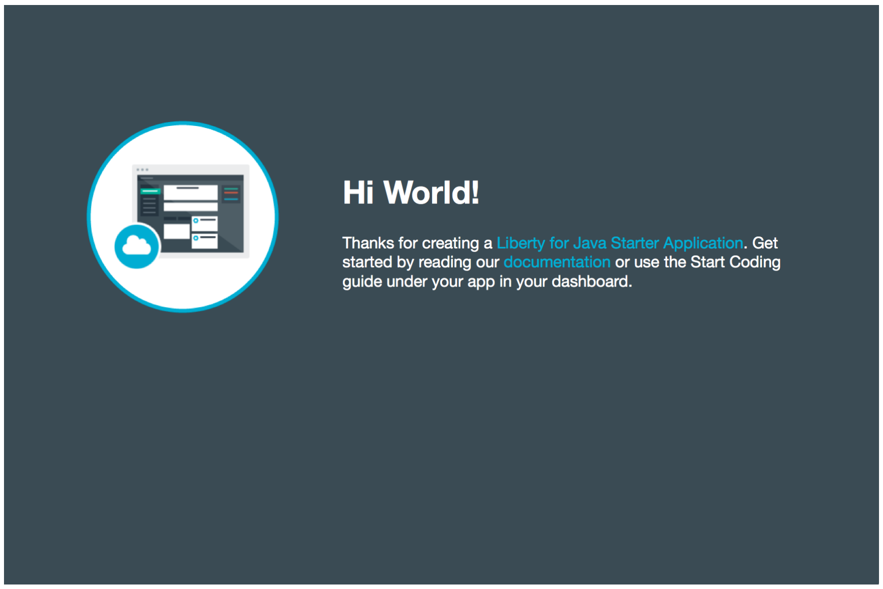


Your app is running.

Now that we have a running web app, we should at least look at it. Click on the ***Routes*** link. It should look something like myfirstapp88.mybluemix.net.



The app will be opened and displayed in a separate browser tab.



Hi World!

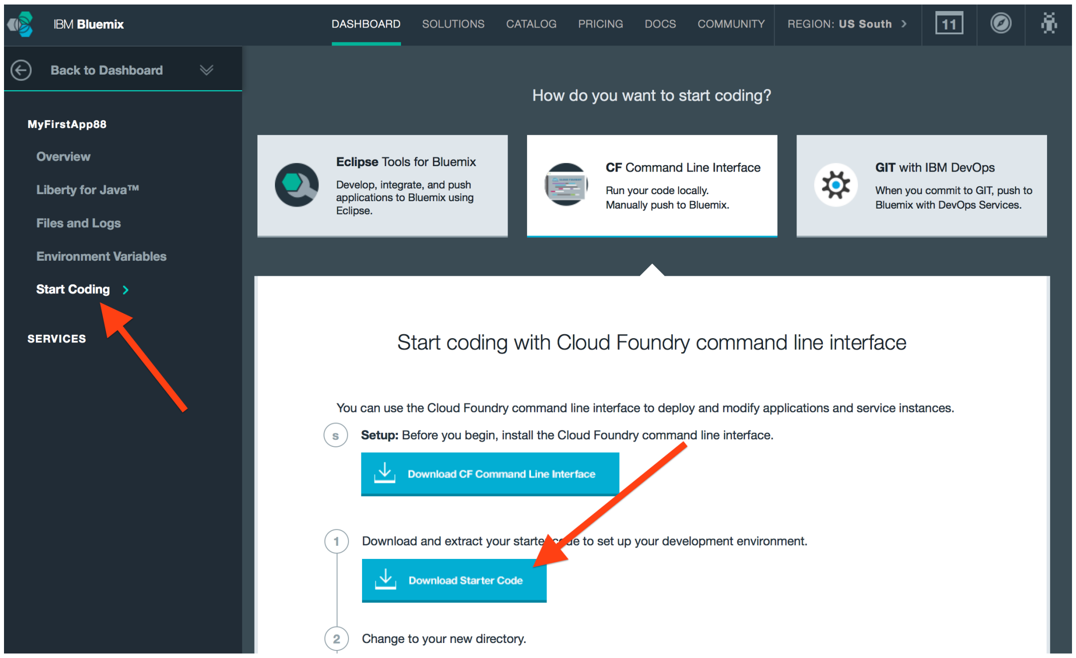
Thanks for creating a Liberty for Java Starter Application. Get started by reading our documentation or use the Start Coding guide under your app in your dashboard.

There is also a simple REST API that was deployed as part of the app.

http://myfirstapp88.mybluemix.net/api/hello

There is not much to our app but it serves as a skeleton that we can build upon. So what does the deployed code look like? We are going to answer that in later labs but you can download it now if you want to have a quick look.

To download the skeleton code, click on ***Start Coding*** and then click on ***Download Starter Code.***



How do you want to start coding?

Eclipse Tools for Bluemix
Develop, integrate, and push applications to Bluemix using Eclipse.

CF Command Line Interface
Run your code locally. Manually push to Bluemix.

GIT with IBM DevOps
When you commit to GIT, push to Bluemix with DevOps Services.

Start coding with Cloud Foundry command line interface

You can use the Cloud Foundry command line interface to deploy and modify applications and service instances.

s **Setup:** Before you begin, install the Cloud Foundry command line interface.

Download CF Command Line Interface

1 Download and extract your starter code to set up your development environment.

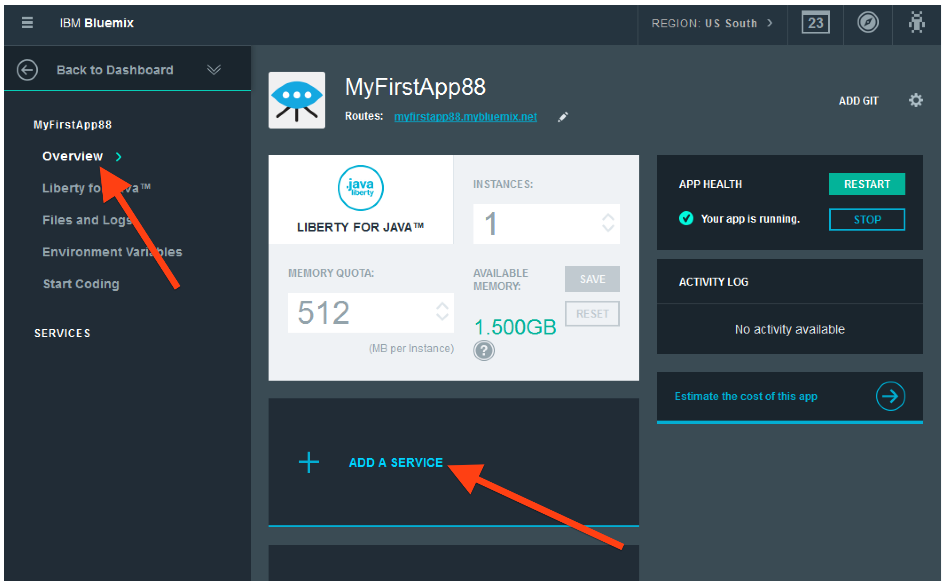
Download Starter Code

2 Change to your new directory.

On the above page, there are full instructions for deploying your code using the command line tools, Bluemix, plugin, and IBM DevOps. We will be covering some of these in the next labs.

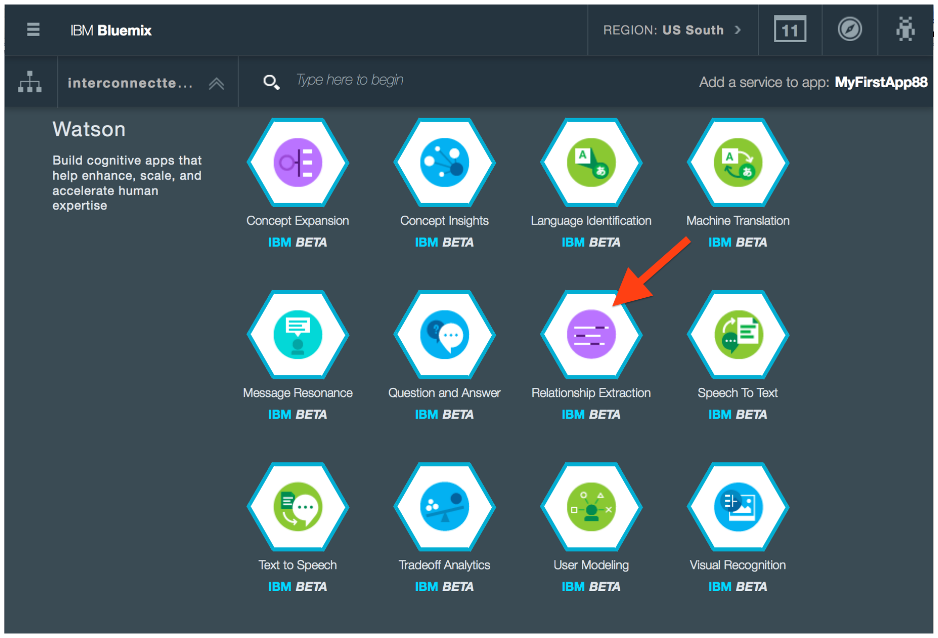
If we just wanted to run a basic app on Bluemix that would be great. Even better is to use one of the many services available on the Bluemix platform. We now want to add a service to our app.

Click on the ***Overview*** link on the left side of the page. Then click on ***Add a Service***.



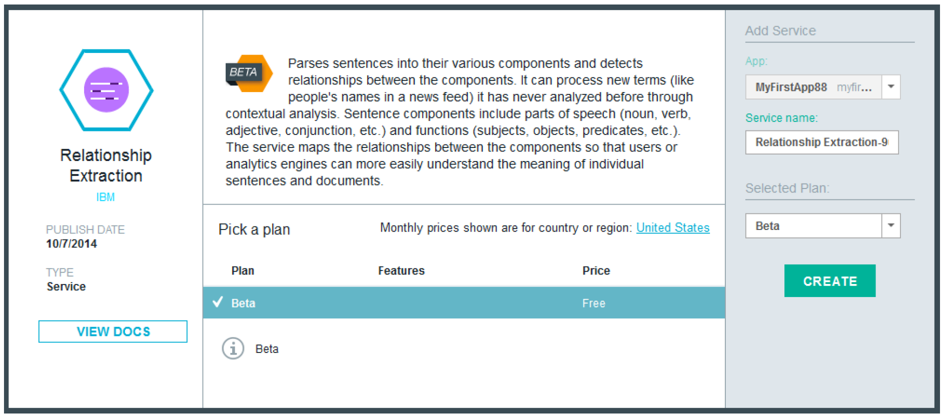
We are now presented with a list of services available through Bluemix. There are many types of services from mobile to big data. For now we are just going to look at a Watson service.

Click on ***Relationship Extraction***.


IBM Bluemix
REGION: US South
interconnecte...
Type here to begin
Add a service to app: MyFirstApp88
Watson
Build cognitive apps that help enhance, scale, and accelerate human expertise
Concept Expansion IBM BETA
Concept Insights IBM BETA
Language Identification IBM BETA
Machine Translation IBM BETA
Message Resonance IBM BETA
Question and Answer IBM BETA
Relationship Extraction IBM BETA
Speech To Text IBM BETA
Text to Speech IBM BETA
Tradeoff Analytics IBM BETA
User Modeling IBM BETA
Visual Recognition IBM BETA

Read the brief description of the Relationship Extraction service to get an understanding as to what it does. If you want to find how to use a service, look at the *Docs* link.

Click the ***Create*** button to accept the defaults.



Relationship Extraction
IBM
PUBLISH DATE 10/7/2014
TYPE Service
VIEW DOCS

BETA Parses sentences into their various components and detects relationships between the components. It can process new terms (like people's names in a news feed) it has never analyzed before through contextual analysis. Sentence components include parts of speech (noun, verb, adjective, conjunction, etc.) and functions (subjects, objects, predicates, etc.). The service maps the relationships between the components so that users or analytics engines can more easily understand the meaning of individual sentences and documents.

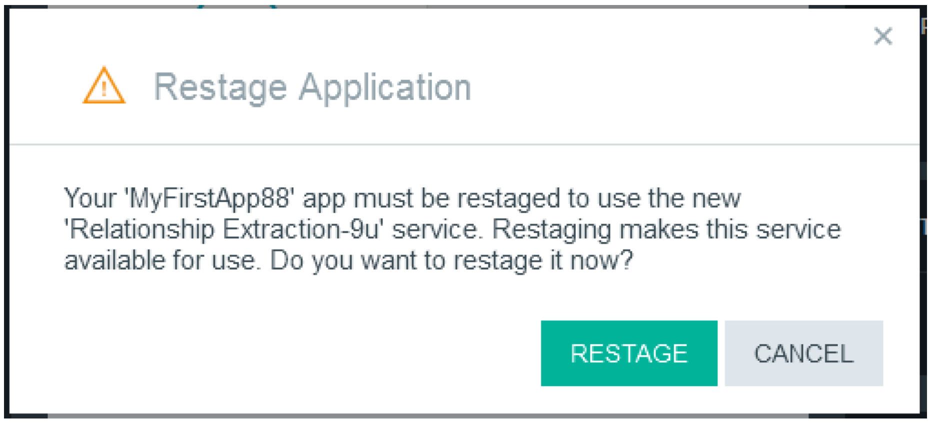
Pick a plan — Monthly prices shown are for country or region: United States

Plan	Features	Price
✓ Beta		Free

Beta

Add Service
App: MyFirstApp88 myfir...
Service name: Relationship Extraction-9
Selected Plan: Beta
CREATE

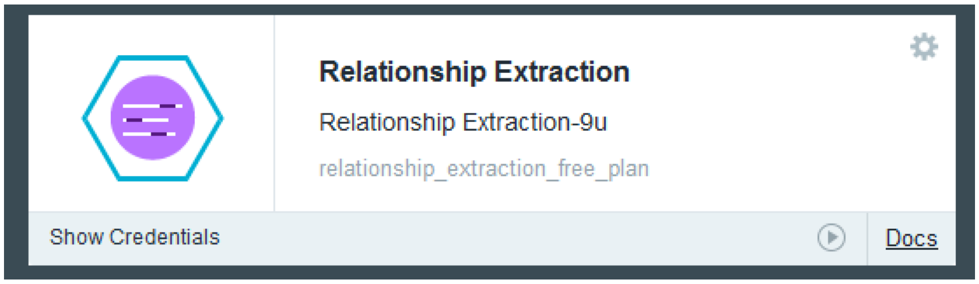
Click ***Restage*** to restart our app and bind the newly added service to your app space.


Restage Application

Your 'MyFirstApp88' app must be restaged to use the new 'Relationship Extraction-9u' service. Restaging makes this service available for use. Do you want to restage it now?

The instance of the Relationship Extraction is bound to your account and has a unique URL and credentials to access it.

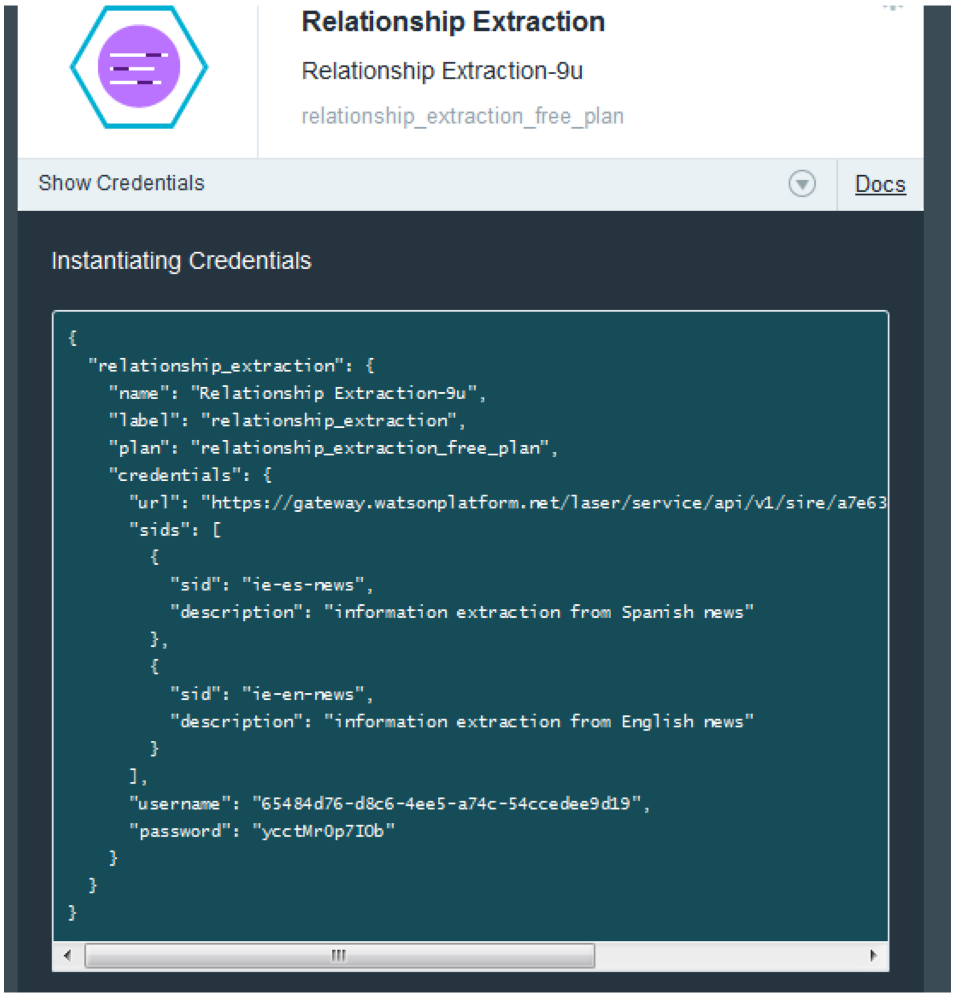
Click on the ***Show Credentials*** to see this connection info.


Relationship Extraction

Relationship Extraction-9u

relationship_extraction_free_plan

Credentials are represented in JSON format. To access them programmatically, you read the environment variable VCAP_SERVICES. VCAP_SERVICES lists all services that are bound to your app.


Relationship Extraction
Relationship Extraction-9u
relationship_extraction_free_plan
Show Credentials
Docs
Instantiating Credentials

```
{
  "relationship_extraction": {
    "name": "Relationship Extraction-9u",
    "label": "relationship_extraction",
    "plan": "relationship_extraction_free_plan",
    "credentials": {
      "url": "https://gateway.watsonplatform.net/laser/service/api/v1/sire/a7e63
      "sids": [
        {
          "sid": "ie-es-news",
          "description": "information extraction from Spanish news"
        },
        {
          "sid": "ie-en-news",
          "description": "information extraction from English news"
        }
      ],
      "username": "65484d76-d8c6-4ee5-a74c-54ccedee9d19",
      "password": "ycctMr0p7IOb"
    }
  }
}
```

In this brief overview we only scratched the surface on what is possible with *Bluemix*. In the following labs we will dig into some code and actually use the *Relationship Extraction* service.