



IBM Bluemix & Watson Labs

Lab 1 - Create & Deploy a Simple Application

Bluemix Console – Dashboard - Service Catalog – Binding – Logs – Service Status

Benoit Marolleau <benoit.marolleau@fr.ibm.com>
Architect – [IBM Montpellier Client Center](#)
IBM France

Labs – Before Starting

- ✓ Access to Bluemix (requires a valid Bluemix Account)
<https://console.ng.bluemix.net>
- ✓ If needed, use a free 30-day trial account
- ✓ Free resources (GB / #Services) in your Bluemix Organization / Spaces to run the lab exercises.
For this hands–on session, you will have to use US-South region to have access to required services (latest versions of services)
If you encounter a resource contention (Error Message saying you are out of resources), clean up your Spaces by deleting existing Apps or Services.
- ✓ Bluemix Public is used for this Lab. Check Bluemix Status & Maintenance schedule on <https://status.ng.bluemix.net/>

Lab 1 – Objective

- Connect to Bluemix, browse the Service Catalog
- Create your first Web Application in Bluemix
- Get familiar with Bluemix Graphical Interface & your Dashboard
- Expected Result: your Web App is operational



Hi World!

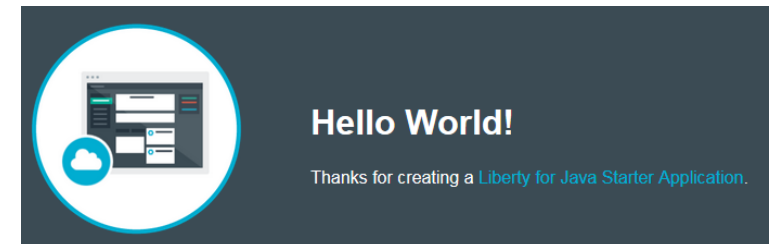
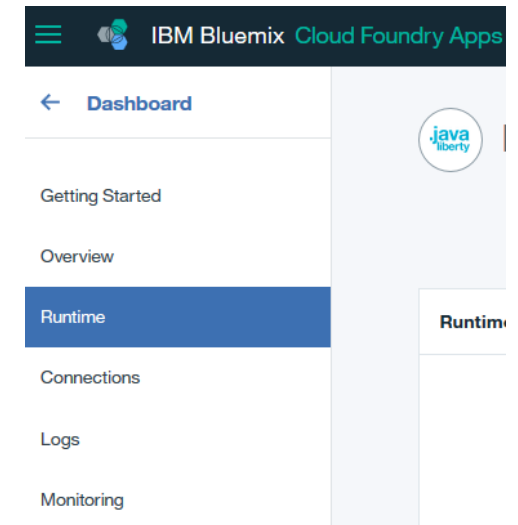
Thanks for creating a [Liberty for Java Starter Application](#). Get started by reading our [documentation](#) or use the Start Coding guide under your app in your dashboard.

Lab 1 – Create your first App

1. Go to Bluemix <https://console.ng.bluemix.net/> & log in
2. See on the top right : Name of your organization & Region (default: US South = IBM Cloud Dallas Datacenter)
3. Go to “Catalog” & scroll down. On the left side, see the categories for a direct & faster access to particular services or runtimes.
4. In the “Cloud Foundry Apps” categories, choose “Liberty for Java” for hosting a Java Enterprise application.
5. Enter the App Name, Host name (part of the domain name to access your app – has to be unique), and click on the CREATE button.
6. Wait for the app to be created. While the container is being created, have a look to the current documentation page: the "Getting Started" page explains how to use the CF (cloud foundry) client to push your application code to Bluemix from your desktop. *It is not part of this lab session. An optional lab is available if you want to test the CF cli or Eclipse CF Plugin.*
7. Your App is now in a “Running” state (green) Click on the “Visit App URL” link.
8. On the previous page (Application Details), go to left side: “Overview”, “Runtime”, “Connections”, “Logs”



BUILDPACK
Liberty for Java™



Lab 1 – Optional: Create & Bind a Service

Note : “Connecting” or binding a [Watson or anything] service to an App is not mandatory to use this Service (a service can run without an application). It is just an elegant way to get the service information instead of hardcoding it in the app.

This information can be used by your application code to access the service. Service information can be: URL, user, password, port, IP. Depending on the type of service. For example, a typical Watson service = your Service URL + Credentials

9. Let’s add a new service & bind it to this application. For example, let's say that newly coded Java application will access a PostgreSQL Database . In “Overview” > “Connections” , “Connect New”

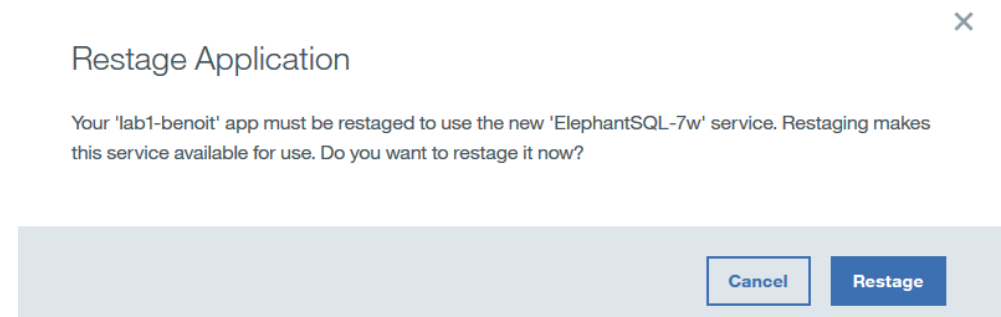
10. Choose the appropriate service. In our case, ElephantSQL (third party “PostgreSQL as a service Service). Name your service, Choose the FREE plan (<20MB of Data) and click CREATE.

This will create your Database service and bind it to your application (The GUI will perform two CF commands for you). Note: Binding ⇔ Create environment variables that your app can use to access the bound services. This creation is done by a “Restage Application”.

11. Restage your application as requested

12. Click on “View Connection” to see the Database service information.

13. Go to your App Dashboard “Runtime” panel > “Environment Variables”.
In our simple “Hello World” App, this does not change anything as we have not modified our code for using this newly created service.



End of Lab

Thank You



Benoît Marolleau

*IBM Certified Experienced Architect
Cloud Computing
Power Systems*



*IBM Client Center Montpellier
Parc Industriel de la Pompignane
34000 Montpellier
Phone: +33 4 67 34 63 35
benoit.marolleau@fr.ibm.com*



To go further...

Reference

- <https://github.com/apischdo/WOW2016>

- IBM Watson Visual Recognition APIs

- Advanced dialog with Conversation....

- More to come...

Watson Developer Cloud (Docs, Demos, Tutorials...)

- <http://www.ibm.com/watson/developercloud/>