

Vaadin Bluemix Boilerplate Introduction

IBM Bluemix Hands-on Workshop

Please enjoy this step-by-step workshop exercise for getting familiar with the Vaadin Boilerplate inside Bluemix. This exercise will guide you through:

1. Creating your own instance of the SimpleCRM example application inside Bluemix
2. Creating your own fork of the codebase
3. Making small modifications to the codebase on-line at hub.jazz.net

After this all the fun is up to you. Happy coding!

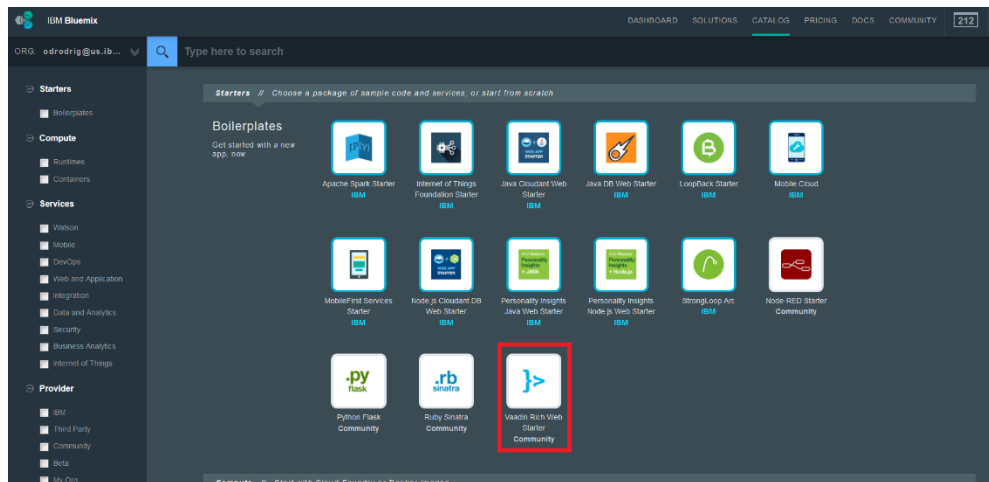
Preparations

1. Make sure you have an account to bluemix.net. If you don't please create one before continuing.
2. Make sure you have an account to hub.jazz.net. If you don't please create one before continuing.

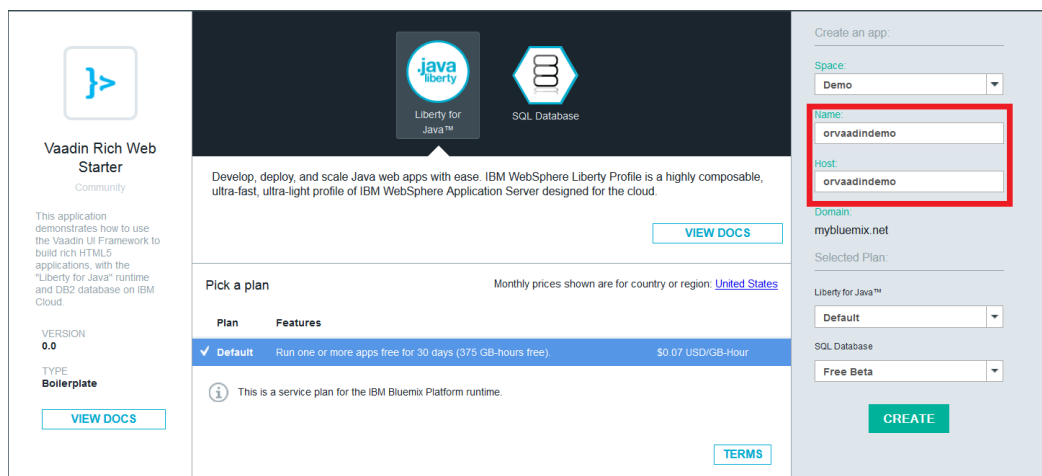
Part #1: Using the on-line tooling

Create your Bluemix instance from the boilerplate

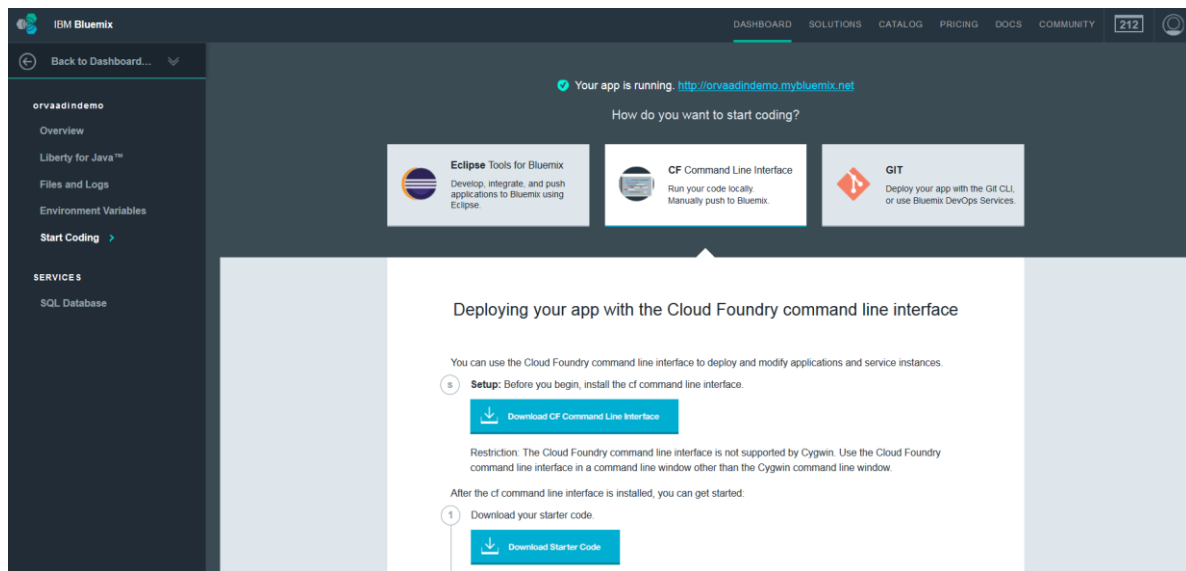
1. Login to bluemix.net with your IBM Id
2. Go to Bluemix catalog page and click the Vaadin Boilerplate, see below



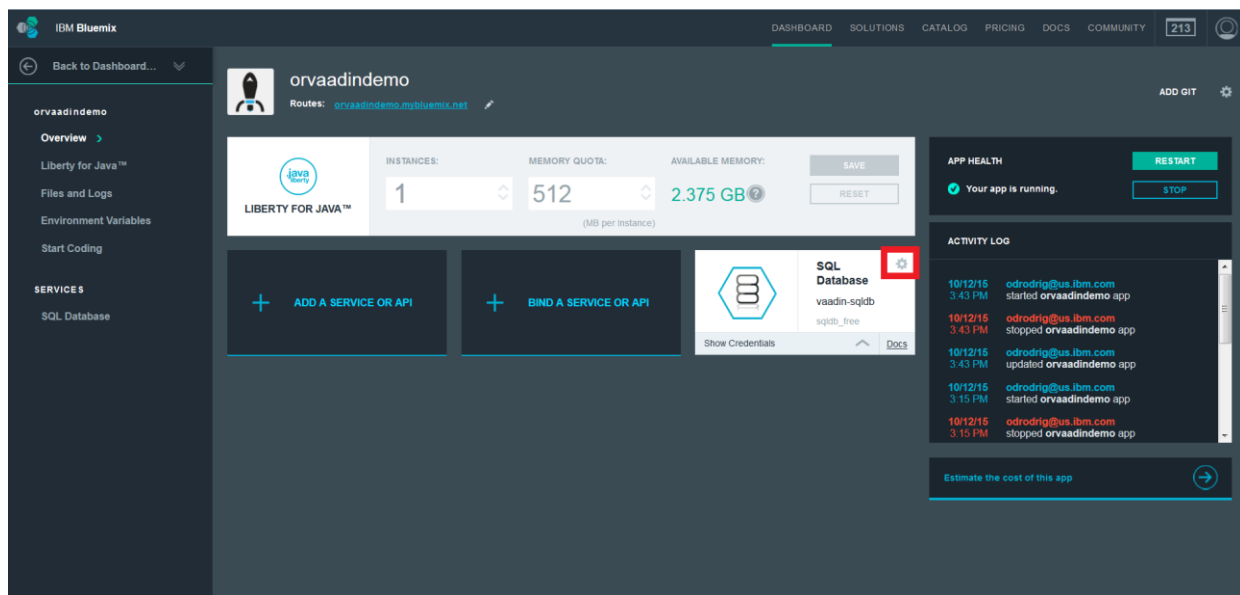
3. Enter a unique 'name' and 'host' for your application, I'm using "orvaadindemo". Avoid spaces and special characters. Once ready click "Create".



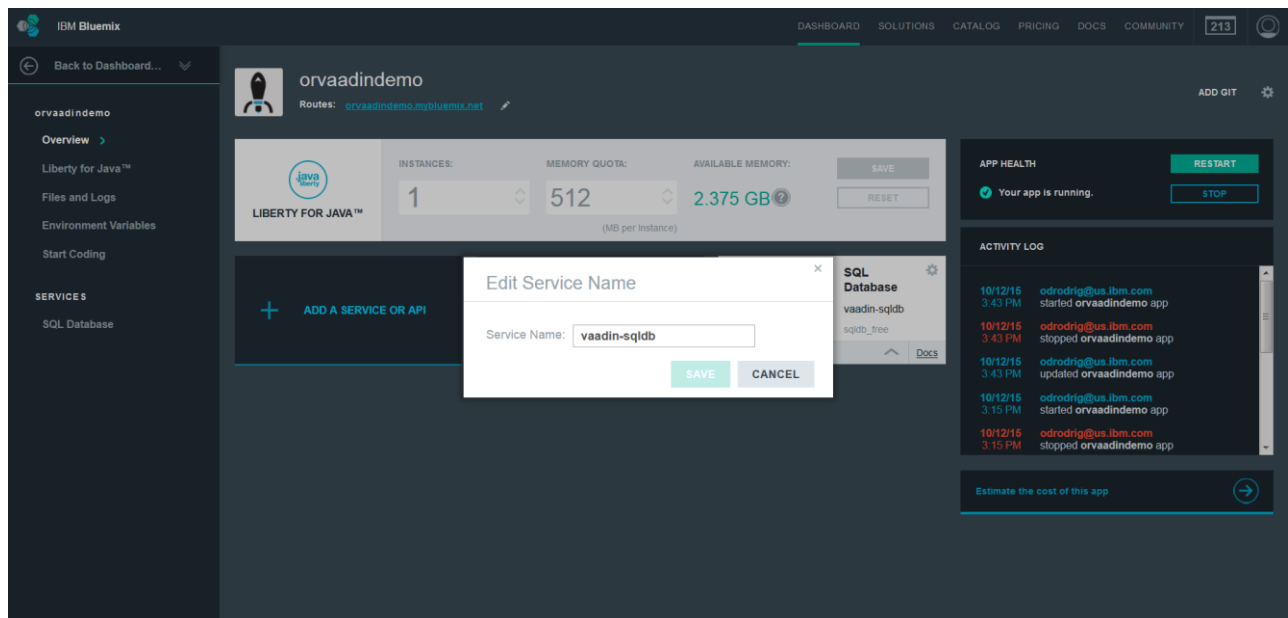
4. Now your application instance is being created so sit back and wait for a while. When everything is ready you should see a similar page as below. Now you have an empty JavaEE server and a database ready.



- Before we move on, we need to rename the SQL Database service in our application overview. Click on the “Overview” tab on the left side of the page to open up the application overview. Then Click on the “Menu” button in the top right of the SQL Database service box as seen below.



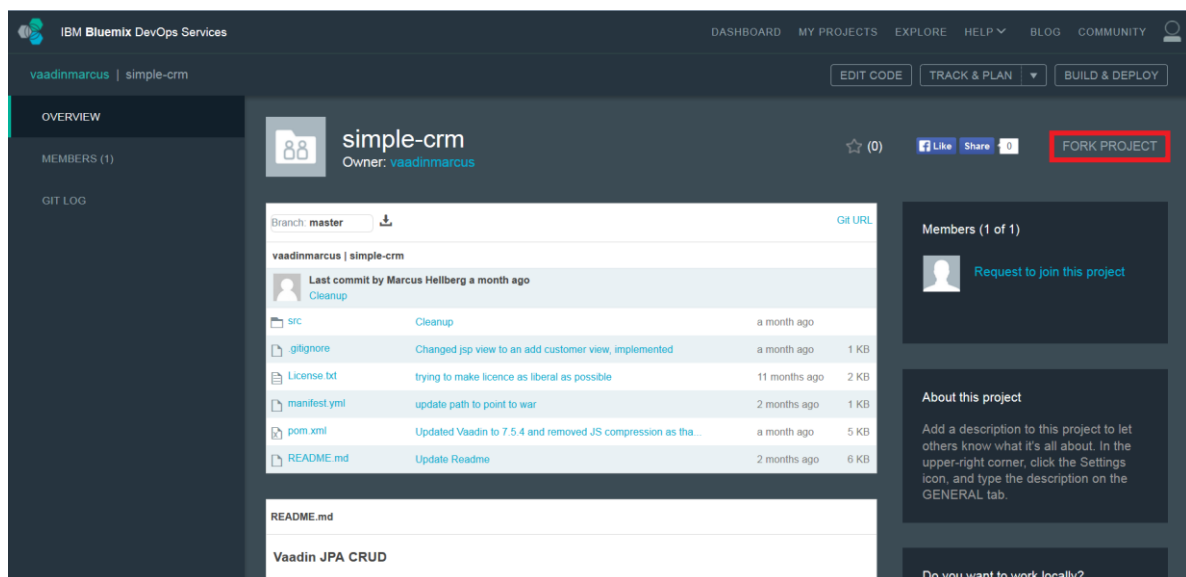
- Select “Rename Service” from the menu and rename the service to “vaadin-sqldb”. The sample code is going to be looking for a service named “vaadin-sqldb” when deployed and if the code cannot find the service, then deployment will fail.



Now that the service has been renamed, we can move on to the sample code.

Setting up Git and deploying to Bluemix

1. Browse to the boilerplate source code repository at <https://hub.jazz.net/project/vaadinmarcus/simple-crm> and click “Fork project”.



2. Configure the forked project

1. Enter the same name as the Bluemix instance that you created earlier.
2. Ensure that the “Make this a Bluemix Project” checkbox is selected.

Fork Project

Name your project: **1**

URL: <https://hub.jazz.net/project/odrodrig/orvaadindemo>

☒ Private project (Invited team members only)

☐ Restrict membership (IBM only)

You can restrict this project's membership because your email address ends with ibm.com. If this project is for **IBM confidential** business, you must select this option and agree to certain conditions. [Learn more](#)

☐ I accept the terms and conditions

☒ Add features for Scrum development ⓘ

☒ Make this a Bluemix Project ⓘ 2

Select a Bluemix space to bill your services to:

Region:

Organization:

Space:

These selections can be changed later in the options for your Project

- Click create and wait while everything is being created for you. When it is finished cloning, you should see the project overview as seen below. This page lists all of the files in the project and the project readme if your project contains one. There is a change we have to make in the code so click on the “Edit Code” Button at the top.

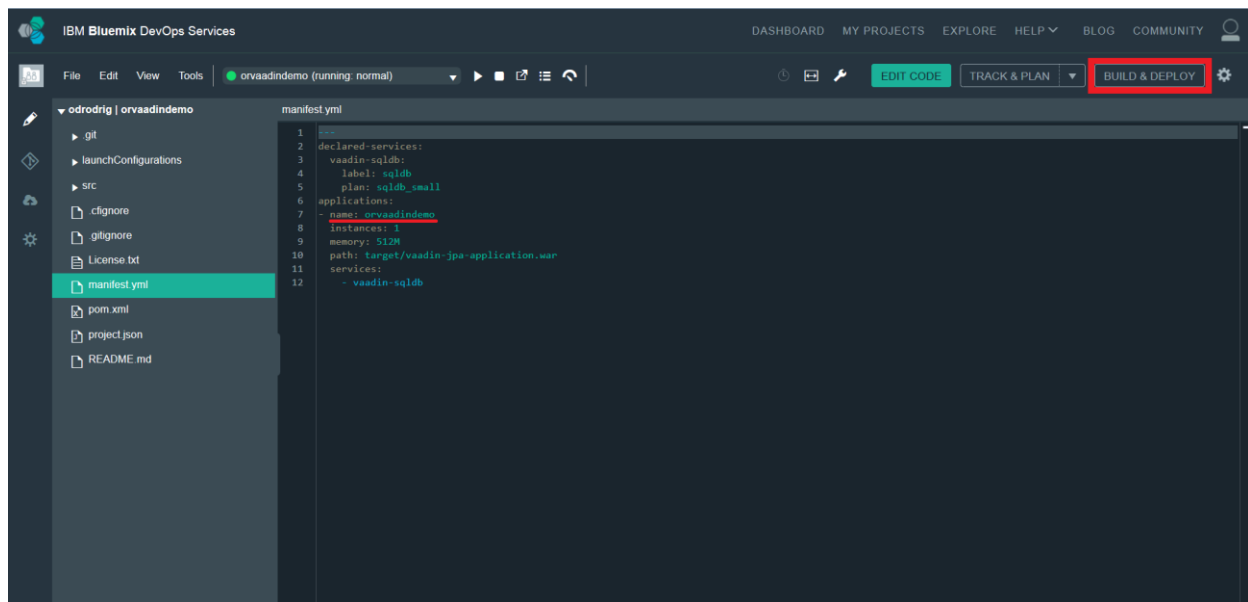
The screenshot shows the IBM Bluemix DevOps Services interface. At the top, there's a navigation bar with 'DASHBOARD', 'MY PROJECTS', 'EXPLORE', 'HELP', 'BLOG', and 'COMMUNITY'. Below this, the project name 'orvaadindemo' is displayed, along with the owner 'odrodrig'. A red box highlights the 'EDIT CODE' button in the top right corner. The main content area shows a list of files and their commit history:

File	Commit Message	Time	Size
src	Cleanup	a month ago	
gitignore	Changed jsp view to an add customer view, implemented	a month ago	1 KB
License.txt	trying to make licence as liberal as possible	11 months ago	2 KB
manifest.yml	update path to point to war	2 months ago	1 KB
pom.xml	Updated Vaadin to 7.5.4 and removed JS compression as tha...	a month ago	5 KB
README.md	Update Readme	2 months ago	6 KB

Below the file list, the README content is visible, starting with 'Vaadin JPA CRUD'. On the right side, there's a 'Members (1 of 1)' section with a user profile and an 'Invite others to join your project' button. Below that, there's an 'About this project' section with a description and a 'FORK PROJECT' button.

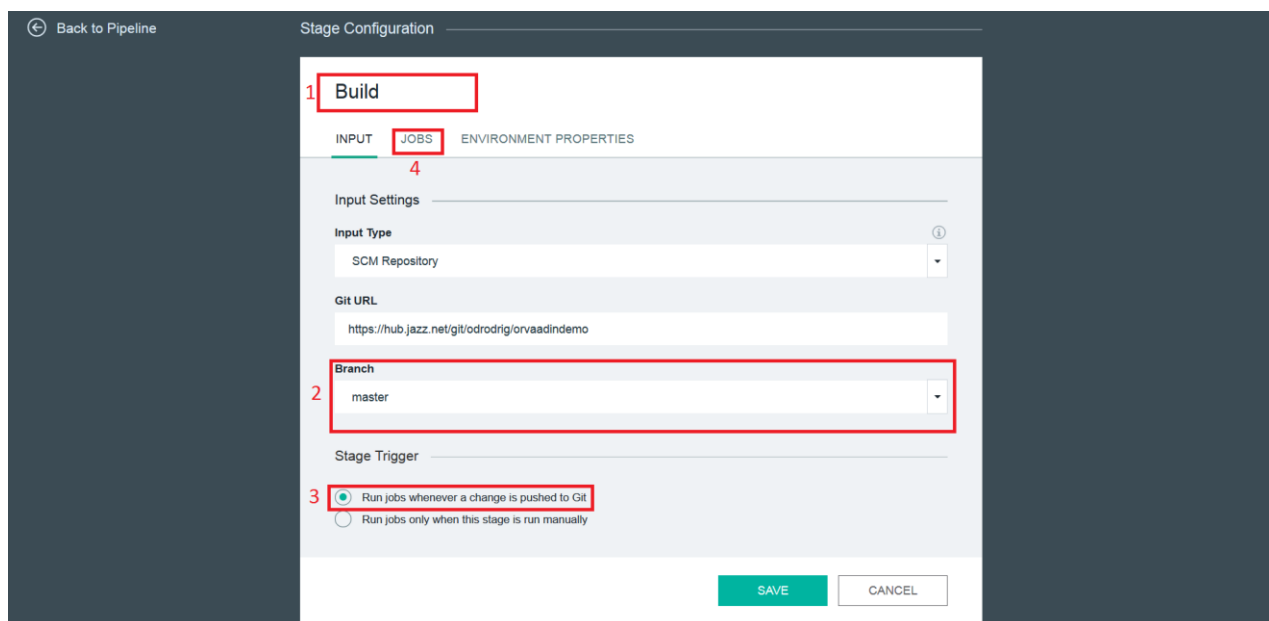
- This will bring you into the web-based IDE where you can make changes to your code directly in the browser. Your project directory is displayed on the left side of the page. Navigate to the “manifest.yml” file. This file is used by Bluemix to configure your application when deploying. Change the “name” section to the same name as your instance in Bluemix. When we deploy the application, Bluemix is going to see that an application already exists with this name so it will overwrite our old code with this new sample code. When you are done, click on the “Build & Deploy button at the top.

Tip: Take notice of the “pom.xml” file. This file is used by Maven to build the application and to specify any dependencies needed for the application to run. This will be important during the build stage later on.



5. This will take you to the Pipeline page where build, deploy, and test stages are configured. Click on “Add Stage”.

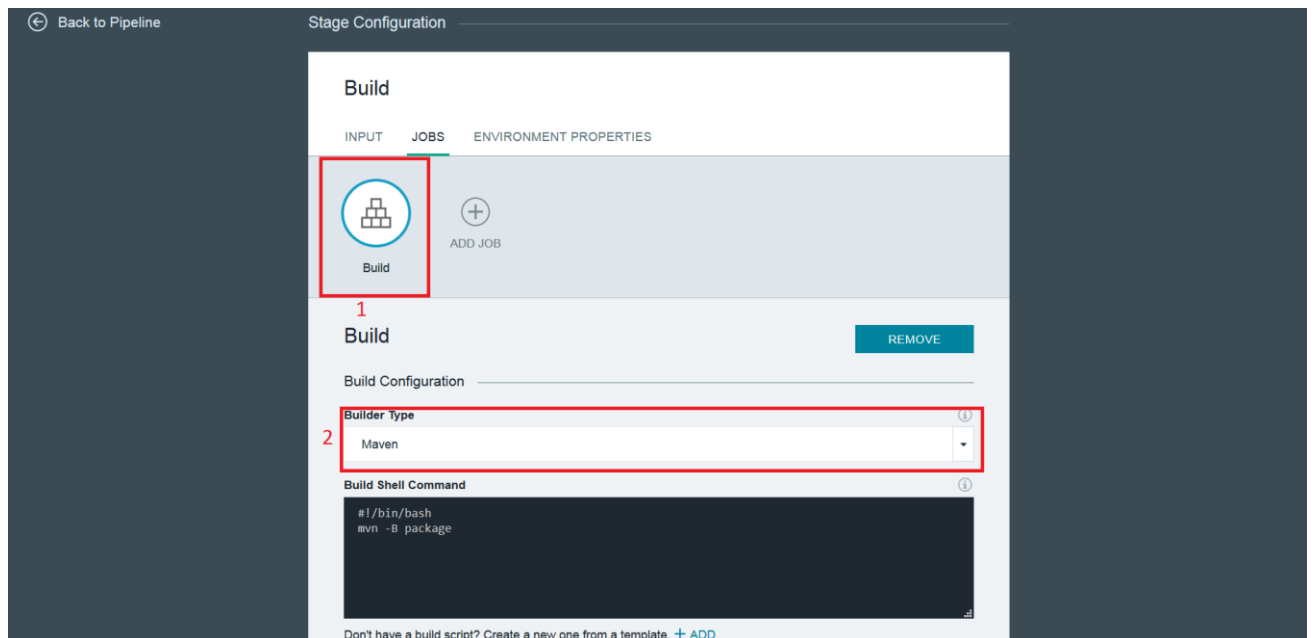
1. Rename the stage to “Build”.
2. Ensure that “Master” is selected under the “Branch” drop down menu.
3. Make sure that “Run jobs whenever a change is pushed to Git” is selected.
4. Click on the “Jobs” tab



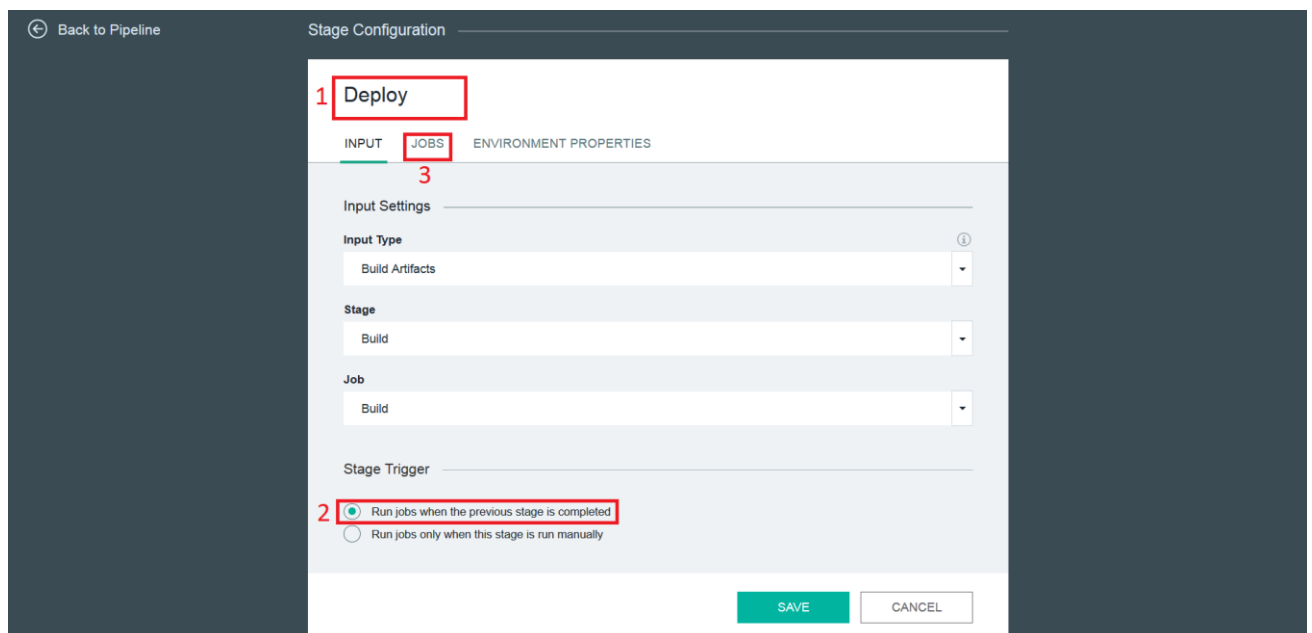
6. Configure the build stage

1. Click “Add job” and select “Build”
2. Under “Builder Type” select “Maven”. When done click “Save”

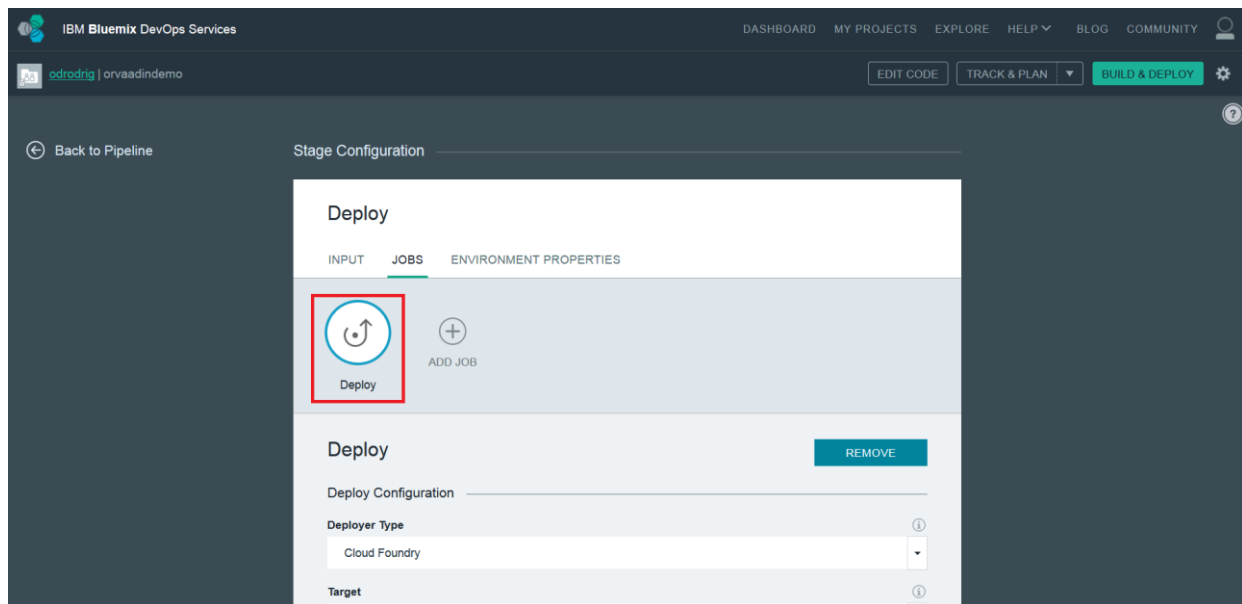
Tip: As mentioned in the tip in step 4, this is a Maven project since it has a pom.xml file associated with it. During this stage, Maven will read the pom.xml file and download the dependencies and package the application in to a .war file.



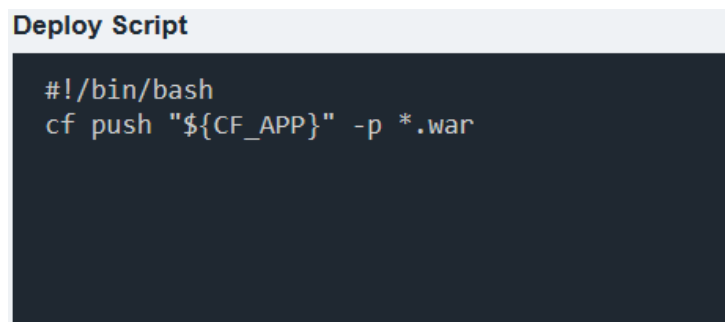
7. Now we need to add a Deploy stage to the pipeline. Click on “Add Stage”
1. Rename the stage to “Deploy”
 2. Ensure that “Run jobs when the previous stage is completed” is selected
 3. Click on the “Jobs” tab



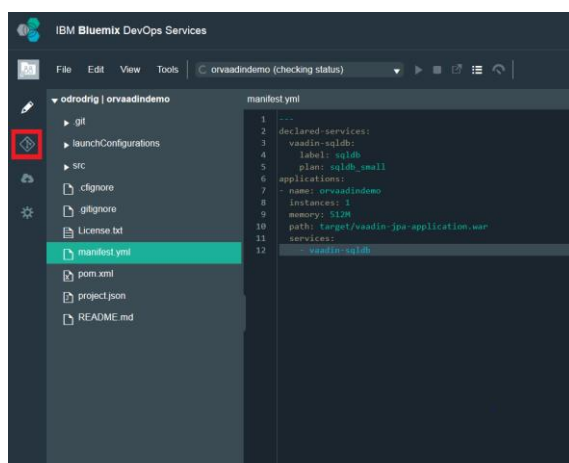
- Click “Add job” and select “Deploy”



- Under the Deploy Script section add “-p *.war” to the end of the command as seen below. This will tell Bluemix to look for any .war file in the project directory. This .war file is created in the build stage prior to the deploy stage.

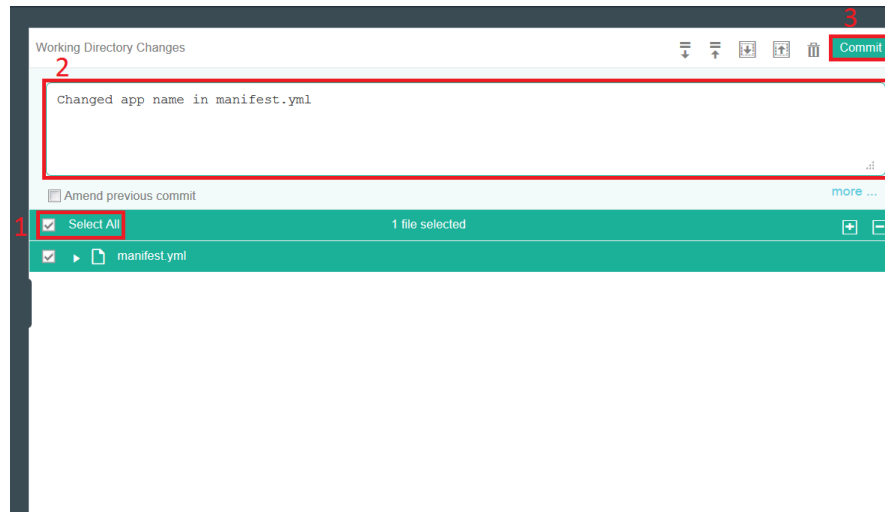


- Click “Save”
- Now we need to push our changes in Git. Click on the “Edit Code” button at the top of the page to take you back to the web-based IDE.
- Then, Click on the Git icon that is under the pencil icon on the left side of the page.

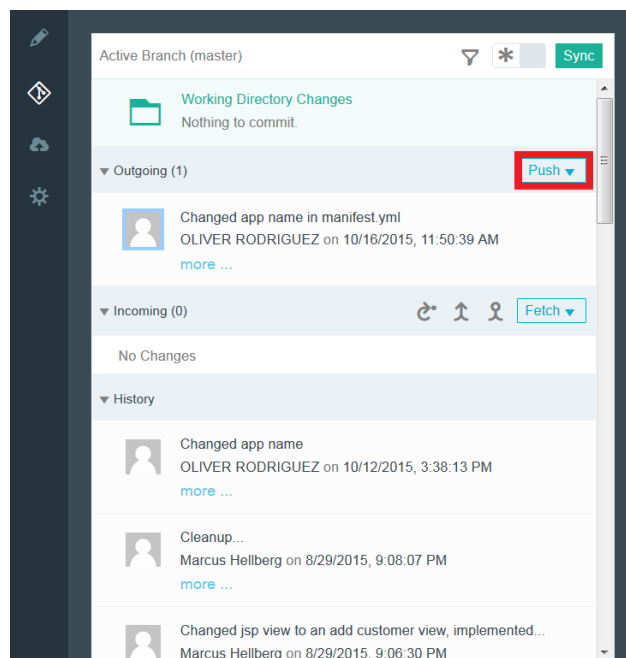


13. This will take you into the Git Repository page where you can commit and push changes.

1. Select the “Select All” check box. This list contains all of the files that have been changed in your project. Selecting them all indicates that you would like to push all of the changes made to the files.
2. Enter a commit message in the text field above the list of files. This message is meant to be descriptive so that anybody looking at the code will know what was changed in this push.
3. Click the “Commit” button to the right of the text field. This will add the commit to the “Outgoing Changes” section to the left.



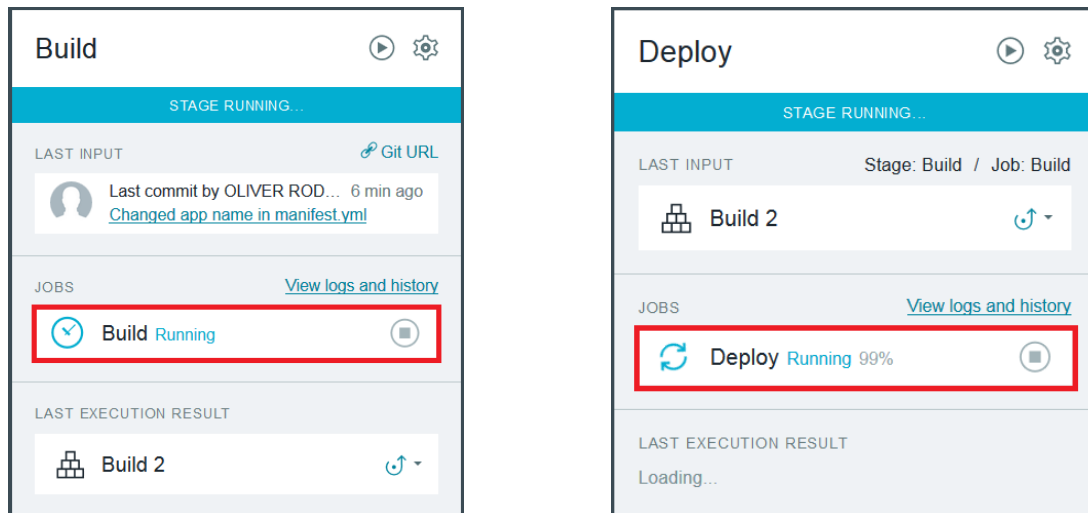
4. Click on the “Push” button. This will update the code in the remote branch and make your changes available to everybody else on the project.



5. After pushing the code, let's take a look at our deployment pipeline in action. Click on the “Build and Deploy” button at the top right.

14. Since we created our deployment pipeline before pushing the changes, our Build stage is now automatically running. After the Build stage completes then the Deploy stage will take the build artifacts and deploy them to Bluemix. Wait while these stages run.

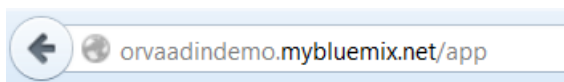
Tip: Optionally, you can click on “Build” or “Deploy” in the “Jobs” box in their respective stage to view the logs. This is important when troubleshooting errors.



15. When your application finishes deploying, go ahead and click on the URL in the “Last Execution Result” to take you to your app.
16. You will be taken to a 404: file not found page. This is because this route is not defined in the sample code that was forked.

Error 404: SRVE0190E: File not found: /

17. You will have to add “/app” to the end of the address bar to reach the actual application



18. This will take you to the running application. You will have to import the sample data into the database. Click on the “Fill test data into DB”.

Simple CRM

Customer List

Analyze

Map

About

Legacy

Simple CRM

This is an example application generated by BlueMix boilerplate.

Features that you get when starting your application from this boilerplate includes:

- The app uses a relational database for persistence, namely DB2 during deployment into BlueMix. Feel free to configure any RDBMS for local development.
- The data is accessed through JPA and EJB using container managed transactions.
- A rich HTML5 UI implemented with Vaadin Framework in plain Java for easy modification and maintenance. For more information and documentation on Vaadin see vaadin.com
- Built in mobile support, with responsive design.

Note, setting up the database schema and test data may take a while on first run.

There are 0 records in the DB.

Fill test data into DB

19. The app is now complete! Explore the application and take note of how Vaadin takes your data and visualizes it in multiple ways including a table, graphs, and as locations on a map.

Simple CRM

Customer List

Analyze

Map

About

Legacy

Customers

Filter customers...

First name	Last name	Email	Status
Alejandro	Macdonald	alejandro@macdonald.com	Contacted
Emily	Stewart	emily@stewart.com	Contacted
Jayden	Jackson	jayden@jackson.com	Contacted
Brian	Robinson	brian@robinson.com	Customer
Yurem	Jackson	yurem@jackson.com	Closedlost
Remington	Andersson	remington@andersson.com	Notcontacted
Gunner	Karlson	gunner@karlson.com	Notcontacted
Lara	Martin	lara@martin.com	Customer
Solomon	Olsen	solomon@olsen.com	Notcontacted
Ann	Andersson	ann@andersson.com	Closedlost
Bernard	Nielsen	bernard@nielsen.com	Closedlost
Gabrielle	Patel	gabrielle@patel.com	Importedlead
Jamar	Olsson	jamar@olsson.com	Closedlost
Koen	Johansen	koen@johansen.com	Importedlead
Israel	Carlsson	israel@carlsson.com	Contacted
Kathleen	Martin	kathleen@martin.com	Notcontacted

Simple CRM

Customer List

Analyze

Map

About

Legacy

Opportunity

Customer analysis

Age distribution

Sales funnel

Gender

Simple CRM

Customer List

Analyze

Map

About

Legacy

Customers on map