



Java Plays : Eclipse Lab 2

Continuous Integration using DevOps

Presented by:

IBM

ECOD

IBM Cloud

AGENDA

- ❑ Purpose of This Lab
- ❑ Step 1: Fork DevOps Services Project
- ❑ Step 2: Configure and Run Build and Deploy Pipeline
- ❑ Step 3: Test App with Postman
- ❑ Step 4: Modify Code in DevOps
- ❑ Step 5: Push to Trigger Pipeline
- ❑ Step 6: Test App with Postman
- ❑ Summary

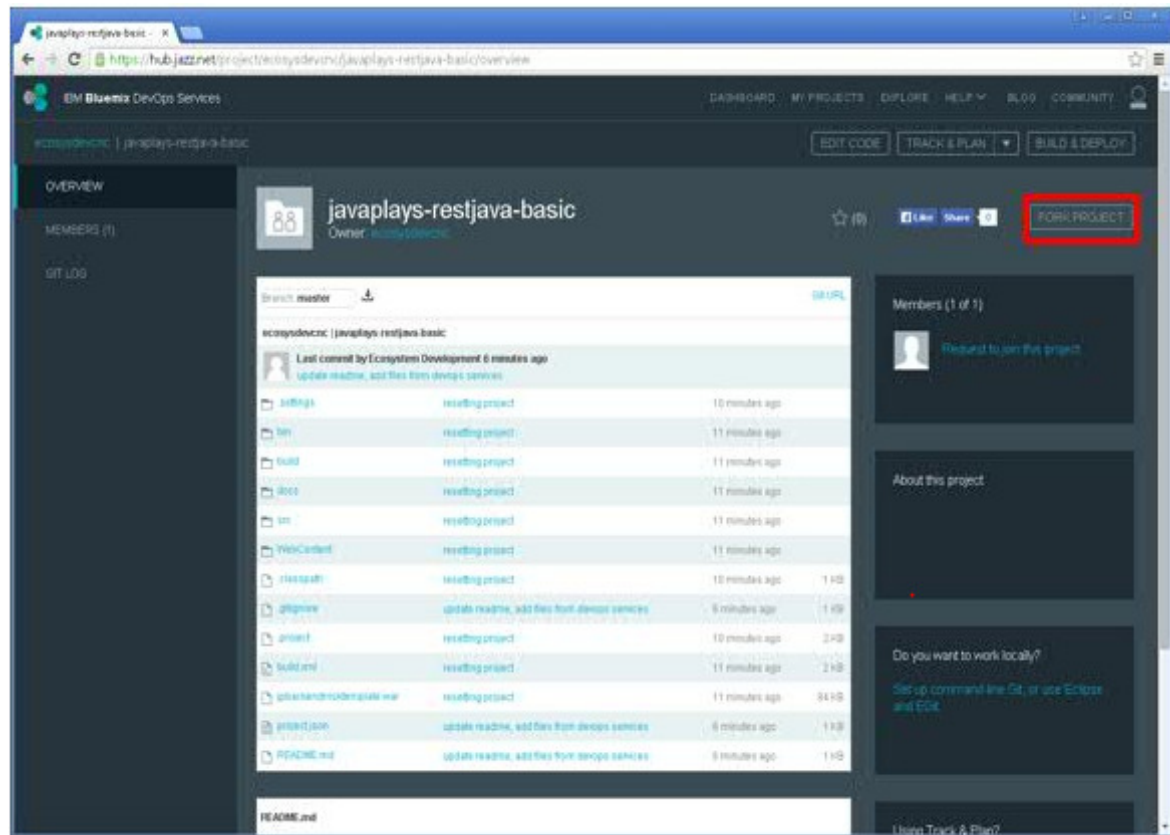
Purpose of This Lab

In this lab, you will learn how to fork a Project in DevOps, edit code and build using DevOps, and finally deploy the project on Bluemix using DevOps. Everything on Bluemix!

Step 1: Fork DevOps Services Project

❑ Browse to **hub.jazz.net**
(<https://hub.jazz.net/project/ecosysdevcnc/javaplays-restjava-basic/overview>)

❑ Click **FORK PROJECT**



Java Plays: Continuous Integration using DevOps

☐ Enter **Project Name**

☐ Check the following checkbox:

- ☐ **Private project**
- ☐ **Add features for Scrum development**
- ☐ **Make this a Bluemix Project**

☐ Select **Region, Organization** and **Space**

☐ Click **CREATE**

Fork Project

Name your project: **holocron** my-restjava-basic ✓

URL: <https://hub.jazz.net/project/holocron/my-restjava-basic>

☒ 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 ⓘ

Select a Bluemix space to bill your services to:

Region: IBM Bluemix US South ▼

Organization: vmorris@us.ibm.com ▼

Space: dev ▼

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

CANCEL CREATE

☐ Success Message



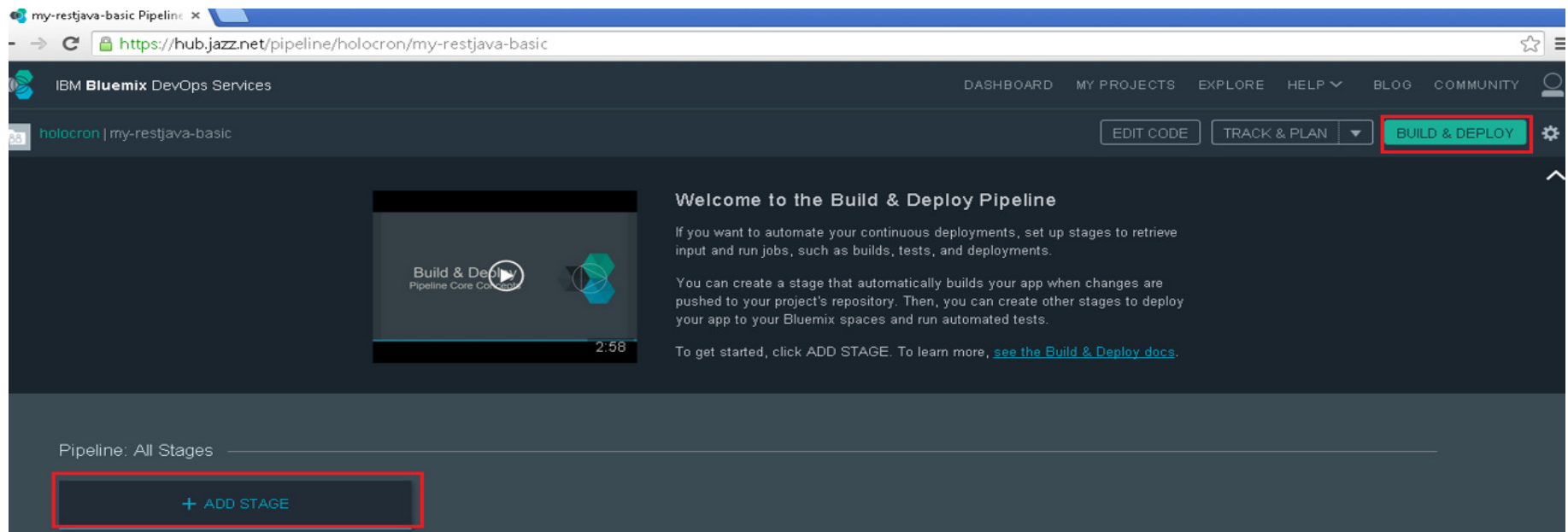
Congratulations! You have successfully created your new project.

Step 2: Configure and Run Build and Deploy Pipeline

To create build and deploy stage

❑ Click **BUILD & DEPLOY**

❑ Click **ADD STAGE**



BUILD STAGE

❑ Enter Stage name

❑ Click **JOBS** tab

Build Stage

INPUT JOBS ENVIRONMENT PROPERTIES

Input Settings

Input Type
SCM Repository

Git URL
https://hub.jazz.net/git/holocron/my-restjava-basic

Branch
master

Stage Trigger

☒ Run jobs whenever a change is pushed to Git
☐ Run jobs only when this stage is run manually

SAVE CANCEL

❑ Click **ADD JOB**

❑ Select **Build**



INPUT JOBS ENVIRONME

SELECT JOB TYPE

Build

Deploy

Test


BUILD STAGE (Continue)


☐ Choose **ANT** for Builder Type

☐ Click **SAVE**

Build Stage

INPUT **JOBS** ENVIRONMENT PROPERTIES


Build


ADD JOB

Build REMOVE

Build Configuration

Builder Type i

Ant ▼

Build Shell Command i

```
#!/bin/bash
ant
```

Don't have a build script? Create a new one from a template. [+ ADD](#)

Working Directory i

Build Archive Directory i

output

☐ Enable Test Report i


Execution Conditions i

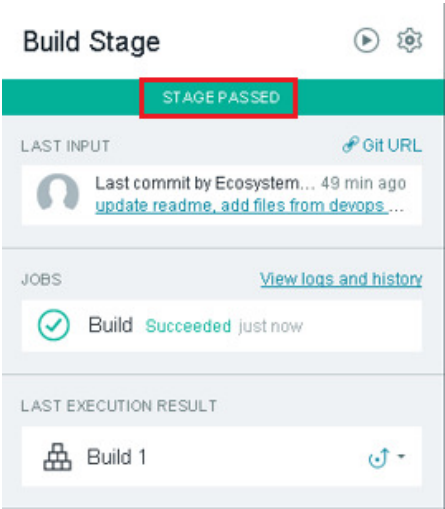
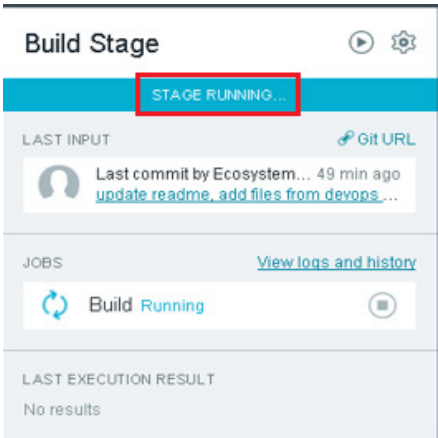
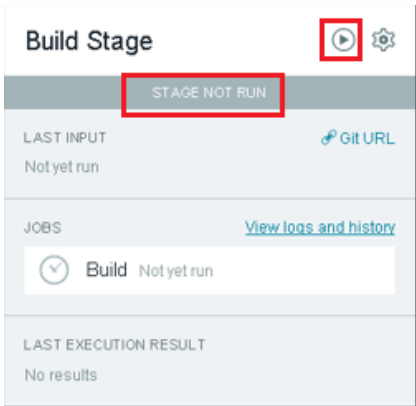
☒ Stop running this stage if this job fails

SAVE

CANCEL

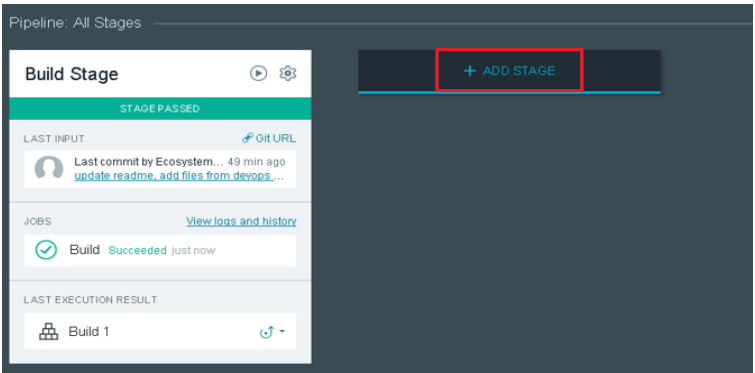
RUN BUILD STAGE

- ❑ See the **Build Stage** at the pipeline
- ❑ Click  (Run Stage Icon)
- ❑ Observe **Stage Status** change from *Stage Not Run* > *Stage Running* > *Stage Passed*



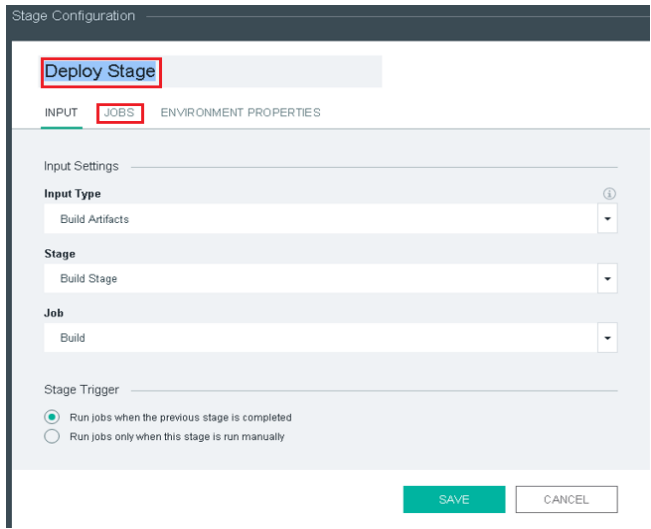
DEPLOY STAGE

❑ At the Pipeline page, click **ADD STAGE**



❑ Enter Stage name

❑ Click **JOBS** tab

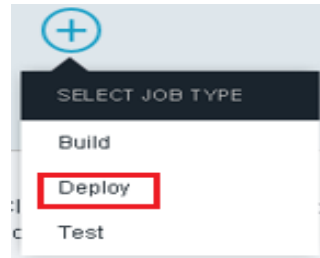


DEPLOY STAGE (Continue)

☐ Click **ADD JOB**



☐ Select **Deploy**



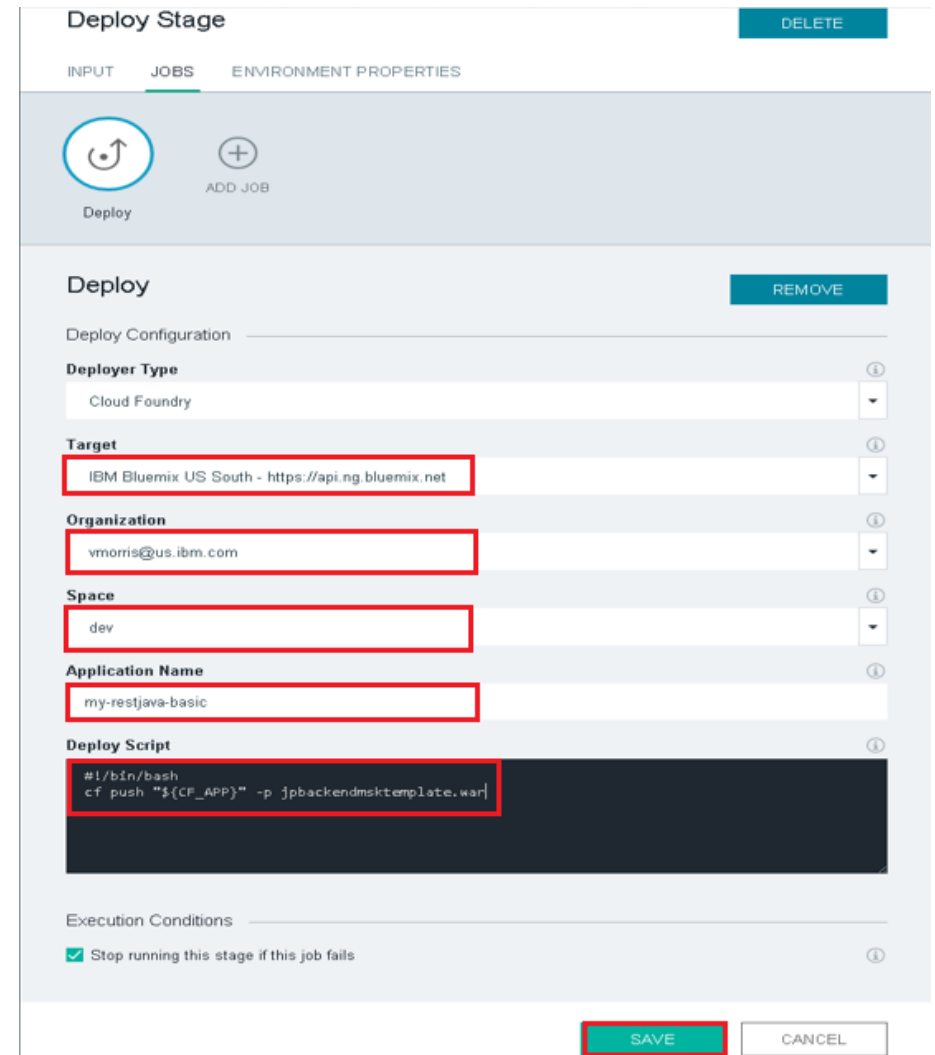
☐ Verify the following

- ☐ **Target**
- ☐ **Organization**
- ☐ **Space**
- ☐ **Application Name**

☐ Edit the Deploy Script


(cf push "\${CF_APP}" -p jpbbackendmsktemplate.war)

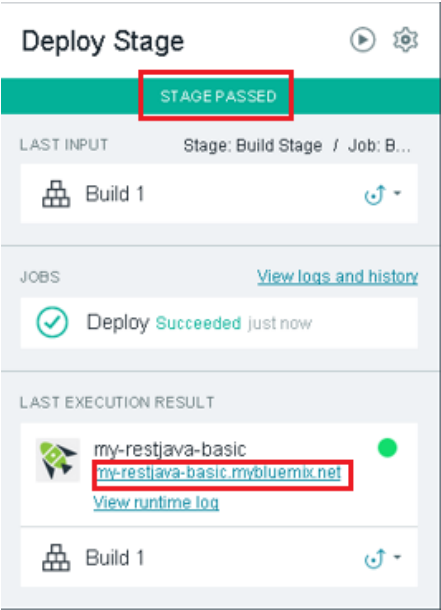
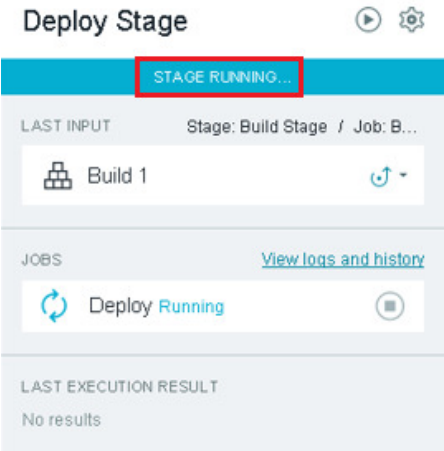
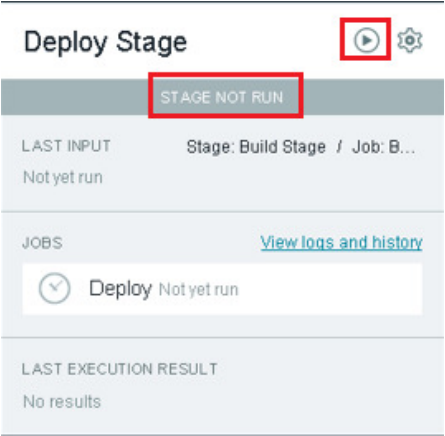
☐ Click **SAVE**

A screenshot of the "Deploy Stage" configuration form in a web application. The form has a header with "Deploy Stage" and a "DELETE" button. Below the header are tabs for "INPUT", "JOBS", and "ENVIRONMENT PROPERTIES". The "JOBS" tab is active, showing a "Deploy" job icon and an "ADD JOB" button. The main form area is titled "Deploy" and has a "REMOVE" button. It contains several fields: "Deploy Configuration" (empty), "Deployer Type" (Cloud Foundry), "Target" (IBM Bluemix US South - https://api.ng.bluemix.net), "Organization" (vmorris@us.ibm.com), "Space" (dev), "Application Name" (my-restjava-basic), and "Deploy Script" (#!/bin/bash, cf push "\${CF_APP}" -p jpbbackendmsktemplate.war). There is also an "Execution Conditions" section with a checked box for "Stop running this stage if this job fails". At the bottom right are "SAVE" and "CANCEL" buttons.

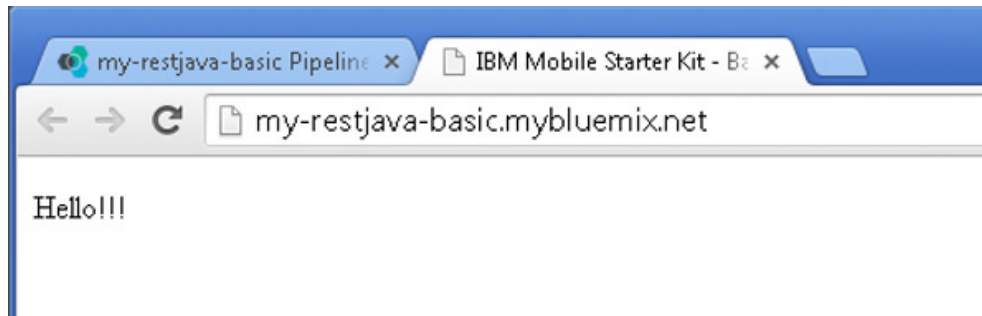
Java Plays: Continuous Integration using DevOps

RUN DEPLOY STAGE

- ❑ See the **Deploy Stage** at the pipeline
- ❑ Click  (Run Stage Icon)
- ❑ Observe **Stage Status** change from *Stage Not Run* > *Stage Running* > *Stage Passed*
- ❑ Click on the link <your hostname>.mybluemix.net to access the deployed application



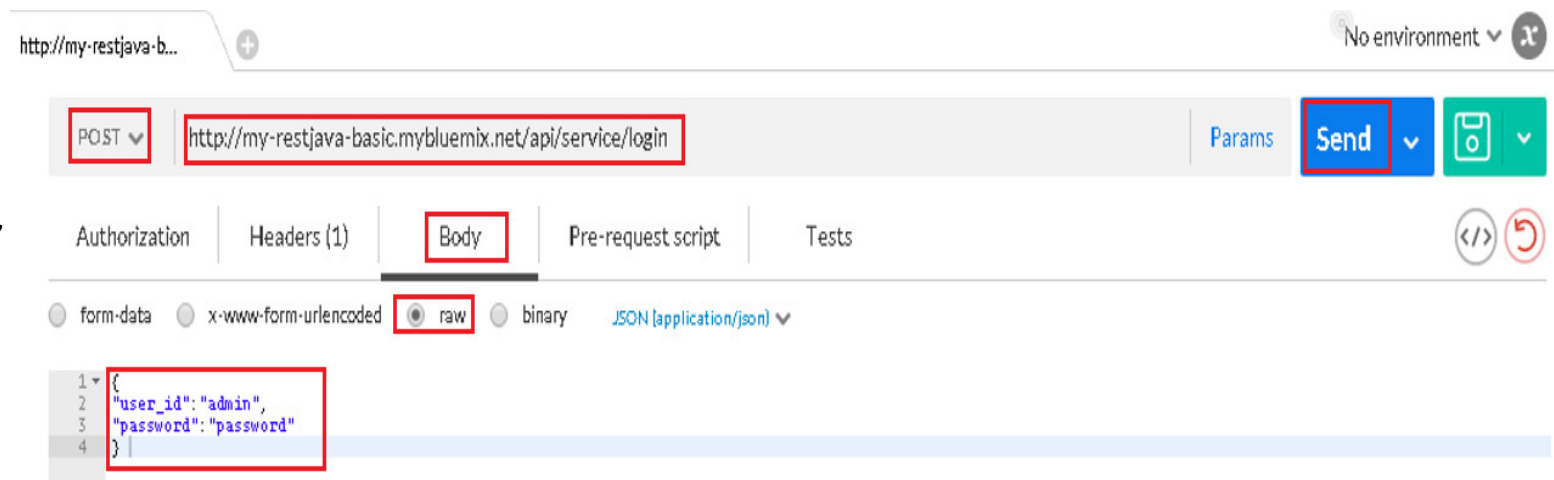
❑ Success page



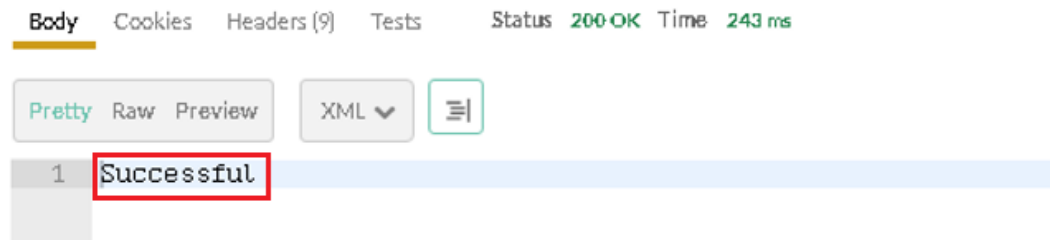
Step 3: Test App with Postman

- ❑ Open **Postman** extension page in Chrome browser
- ❑ Change to **POST** method
- ❑ Enter <http://<your hostname>.mybluemix.net/api/service/login>
- ❑ Click **Body** tab
- ❑ Select **raw**
- ❑ Enter the following

```
{
  "user_id": "admin",
  "password": "password"
}
```
- ❑ Click **Send**



❑ Success message

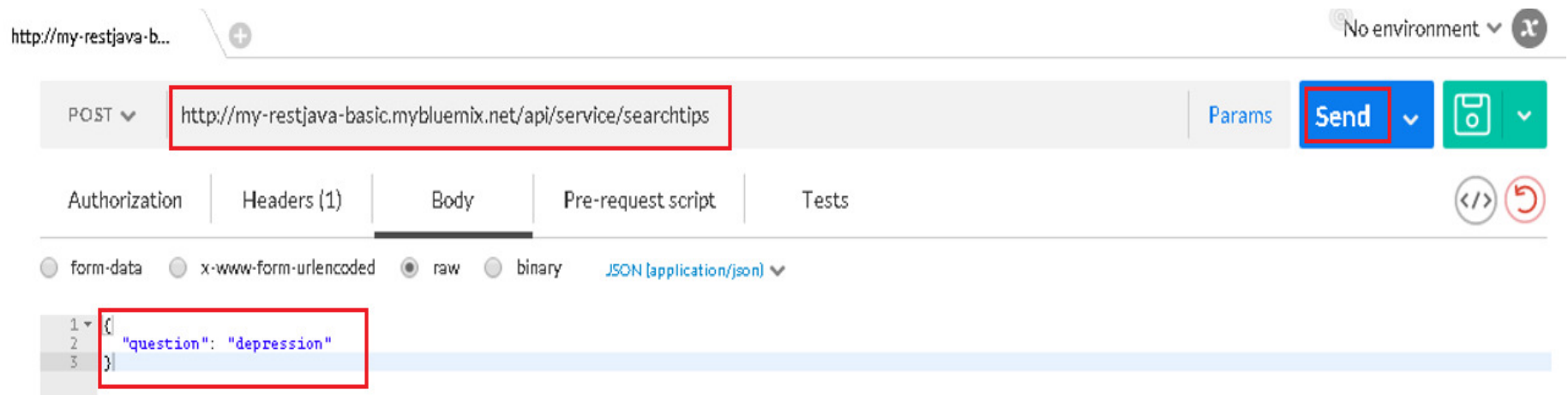


☐ Change url to `http://<your hostname>.mybluemix.net/api/service/searchtips`

☐ Enter the following


```
{  
  "question": "depression"  
}
```

☐ Click **Send**



❑ Success message

Body Cookies Headers (9) Tests Status 200 OK Time 239 ms

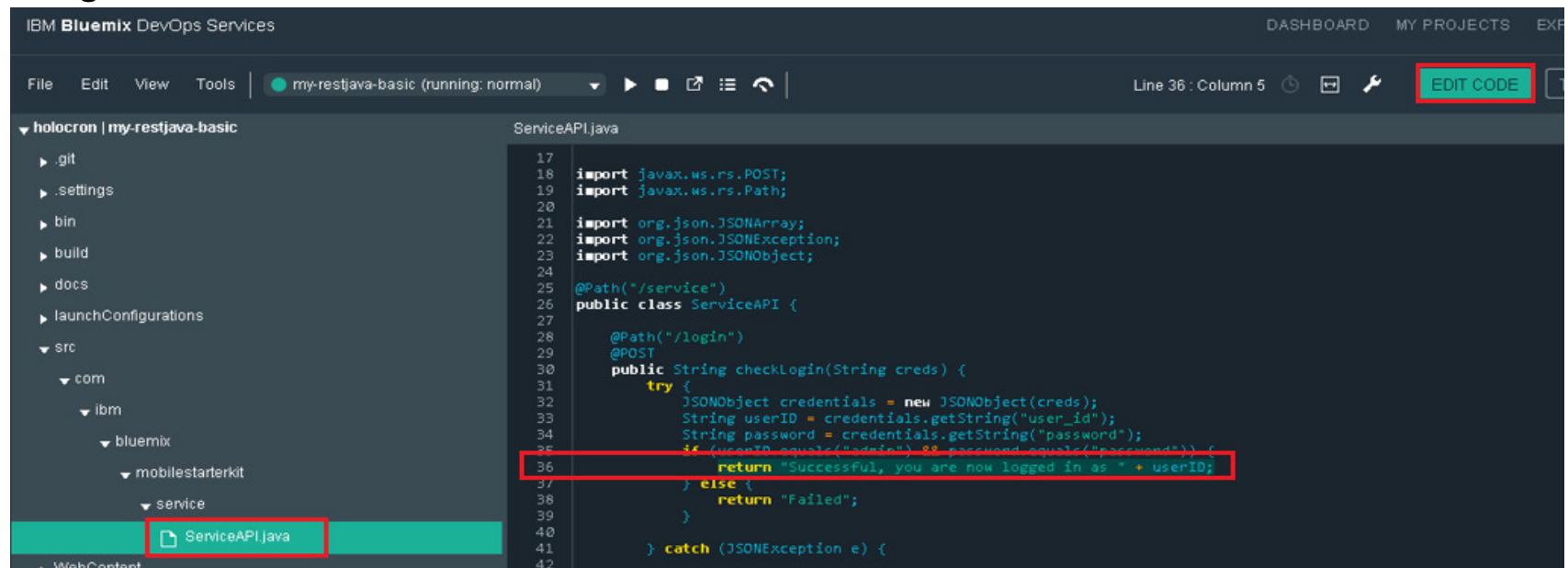
Pretty Raw Preview XML 

1 `{"qa":[{"answers":[{"text":"you can not sleep or you sleep too much"}, {"text":"you can not concentrate or find that previously easy tasks are now difficult"}, {"text":"you feel hopeless and helpless"}, {"text":"you can not control your negative thoughts, no matter how much you try"}, {"text":"you have lost your appetite or you can not stop eating"}, {"text":"you are much more irritable, short-tempered, or aggressive than usual"}, {"text":"you are consuming more alcohol than normal or engaging in other reckless behavior"}, {"text":"you have thoughts that life is not worth living (seek help immediately if this is the case)"}], "question":"What is depression?"}]}`

▼ Scroll to response


Step 4: Modify Code in DevOps

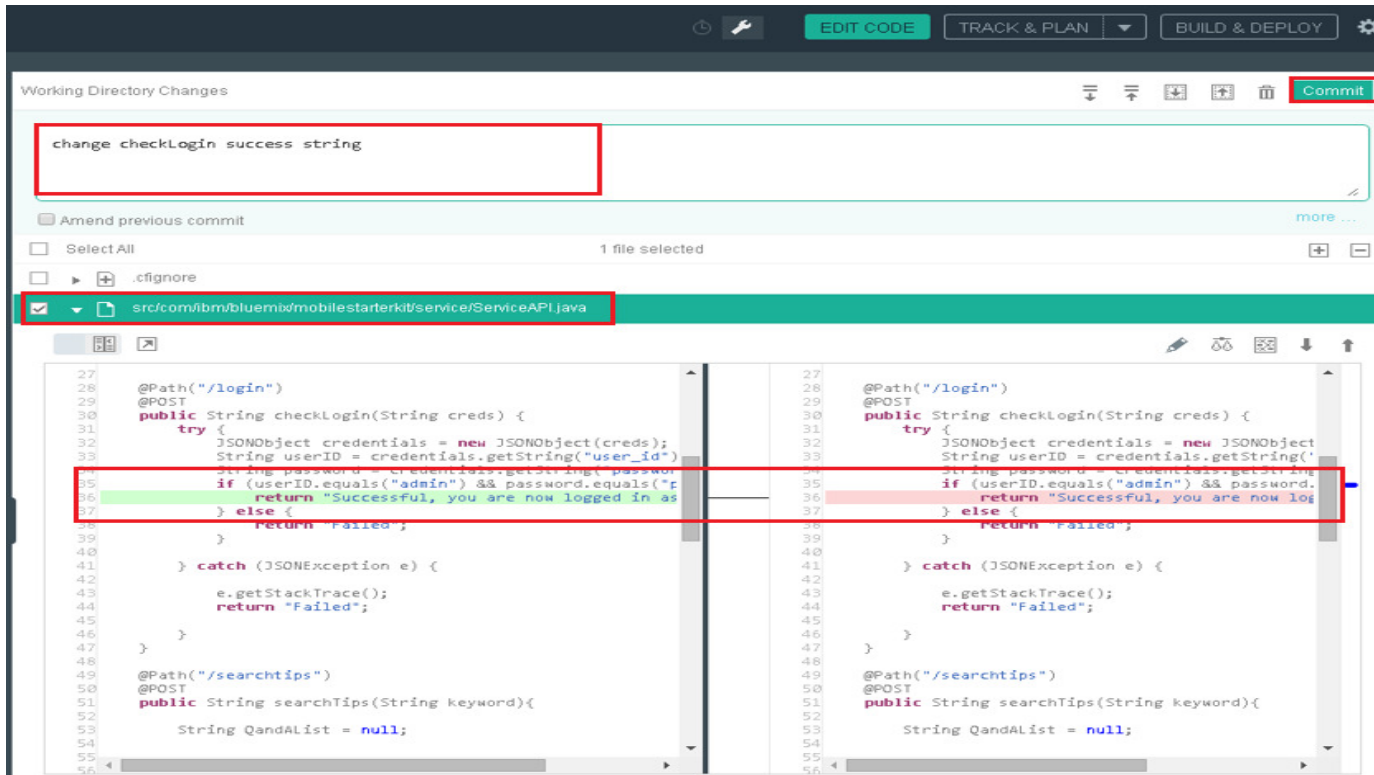
- ❑ Click **EDIT CODE**
- ❑ Go to **src > com > ibm > bluemix > mobilestarterkit > ServiceAPI.java**
- ❑ Edit line 36 to change the success message returned during Postman API POST call
- ❑ **Save** the changes



```
17
18 import javax.ws.rs.POST;
19 import javax.ws.rs.Path;
20
21 import org.json.JSONArray;
22 import org.json.JSONException;
23 import org.json.JSONObject;
24
25 @Path("/service")
26 public class ServiceAPI {
27
28     @Path("/login")
29     @POST
30     public String checkLogin(String creds) {
31         try {
32             JSONObject credentials = new JSONObject(creds);
33             String userID = credentials.getString("user_id");
34             String password = credentials.getString("password");
35             if (userID.equals("basic") && password.equals("password")) {
36                 return "Successful, you are now logged in as " + userID;
37             } else {
38                 return "Failed";
39             }
40         } catch (JSONException e) {
41
42
```

Java Plays: Continuous Integration using DevOps

- ❑ Click **Git Repository** icon 
- ❑ View and compare **before** and **after** changes
- ❑ Add commit message
- ❑ Click **Commit**



Working Directory Changes

change checkLogin success string

Amend previous commit

Select All 1 file selected

src/com/ibm/bluemix/mobilestarterkit/service/ServiceAPI.java

```

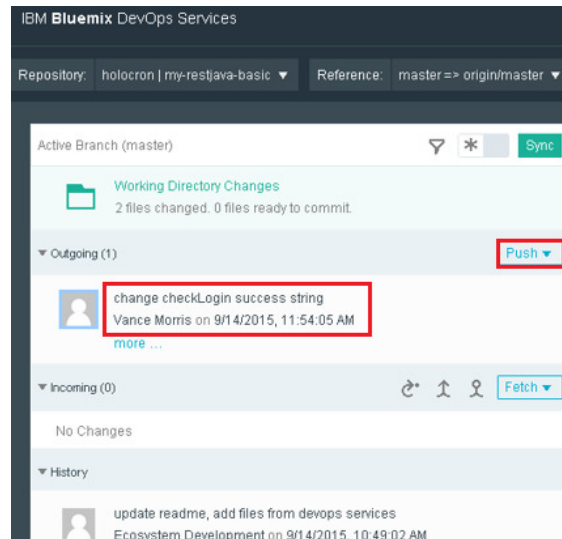
27  @Path("/login")
28  @POST
29  public String checkLogin(String creds) {
30      try {
31          JSONObject credentials = new JSONObject(creds);
32          String userID = credentials.getString("user_id");
33          String password = credentials.getString("password");
34          if (userID.equals("admin") && password.equals("admin")) {
35              return "Successful, you are now logged in as admin";
36          } else {
37              return "Failed";
38          }
39      } catch (JSONException e) {
40          e.printStackTrace();
41          return "Failed";
42      }
43  }
44  @Path("/searchtips")
45  @POST
46  public String searchTips(String keyword){
47      String QandAList = null;
48  }
49  }
50  }
51  }
52  }
53  }
54  }
55  }
56  }

```

Java Plays: Continuous Integration using DevOps

Step 5: Push to Trigger Pipeline

❑ Click **Push**



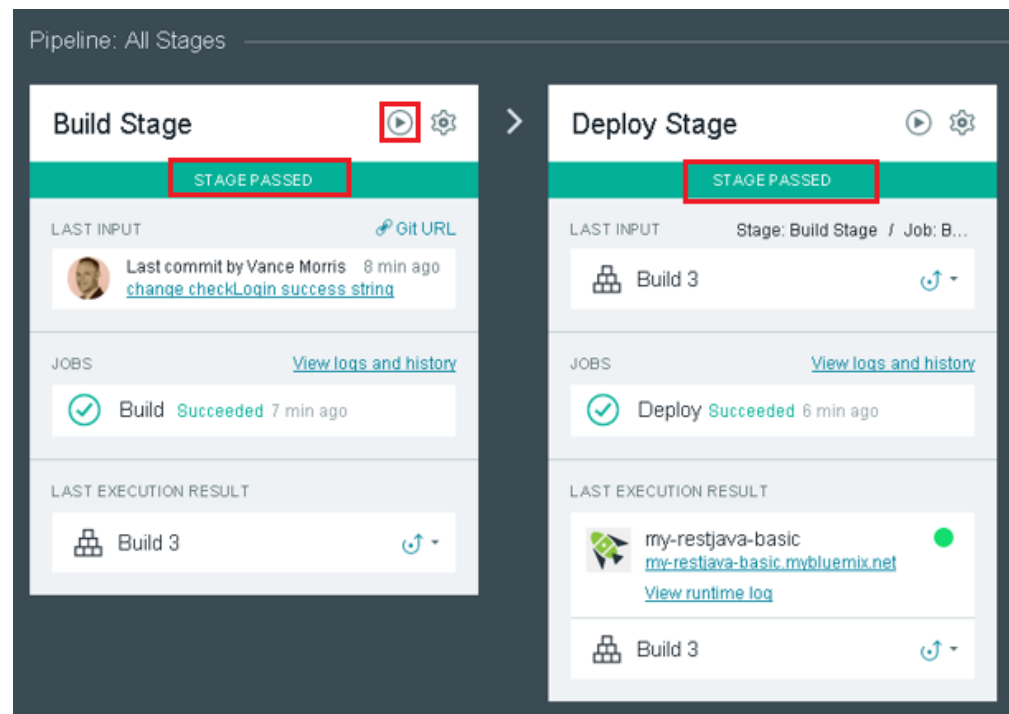
❑ **Success message**

Pushing to remote: origin/master succeeded

❑ Click **BUILD & DEPLOY**

❑ On the **Build Stage**, click  (Run Stage button)

❑ Wait till you see both **Build Stage** and **Deploy Stage** with status **STAGE PASSED**

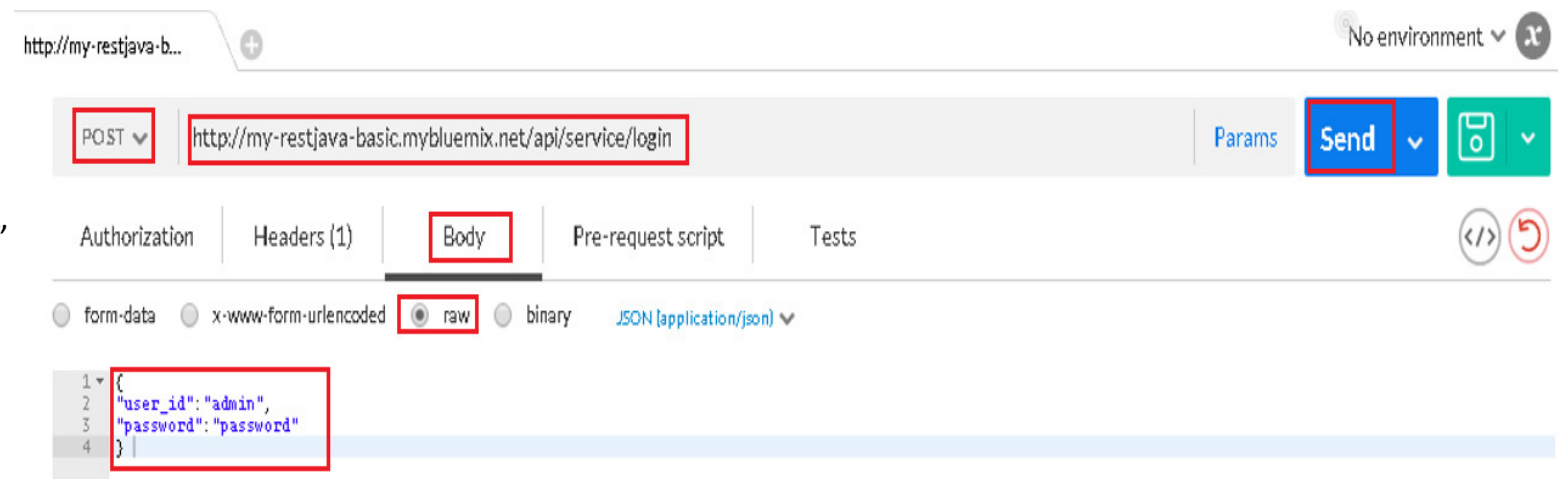


The screenshot displays the Azure DevOps Pipeline interface for a pipeline named "All Stages". It shows two stages: "Build Stage" and "Deploy Stage", both of which have a "STAGE PASSED" status, indicated by a green bar at the top of each stage's card. The "Build Stage" card shows the last input as a commit by Vance Morris 8 minutes ago, and the last job as "Build Succeeded 7 min ago". The "Deploy Stage" card shows the last input as "Stage: Build Stage / Job: B...", and the last job as "Deploy Succeeded 6 min ago". Both stages have a "View logs and history" link. The "Deploy Stage" also shows a "LAST EXECUTION RESULT" for a job named "my-restjava-basic" with a green status indicator and a "View runtime log" link.

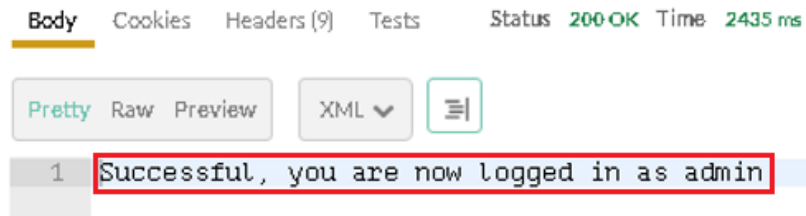
Step 6: Test Modified App with Postman

- ❑ Open **Postman** extension page in Chrome browser
- ❑ Change to **POST** method
- ❑ Enter <http://<your hostname>.mybluemix.net/api/service/login>
- ❑ Click **Body** tab
- ❑ Select **raw**
- ❑ Enter the following

```
{
  "user_id": "admin",
  "password": "password"
}
```
- ❑ Click **Send**



❑ Success message



Summary

In this lab, you learned how to fork a Bluemix project from an existing one, to create the build and deploy stage, to edit the project using DevOps service, to commit and push the changes.

The entire Java application development and deployment cycle can be implemented on a single platform!

This ends the Eclipse Lab 2 Build and Deploy Java Application on Bluemix Server.

Thank you !