08

**Fall**

**Deploy Java Apps on Bluemix and set Up Devops**

Deepak Rai

Table of Contents

Develop and deploy a Java app 3

Before you begin 3

Create an app in Bluemix 3

Create the app 3

Create a Git repository and enable auto-deploy 6

Start using DevOps Services 7

Push changes to the Git repository by using the command line 8

Explore builds and deployments 9

Explore the preconfigured stages 10

Explore the build stage 10

Explore the deploy stage 11

Summary 13

# Develop and deploy a Java app

IBM® Bluemix® is an open-standards, cloud-based platform where you can build, manage, and run all types of apps: mobile, smart devices, web, and big data. The Bluemix capabilities include Java™, mobile back-end development, app monitoring, and features from ecosystem partners and open source, all through an as-a-service model in the cloud.

In this tutorial, you create an app in Bluemix by using the Liberty for Java™ starter pack. You then create a Git repository for the app and add the IBM Continuous Delivery Pipeline for Bluemix (the Delivery Pipeline service). After you change your code, you push the changes to the Git repo from the command line. Then, you build and deploy your app by using the Delivery Pipeline service.

# Before you begin

To do this tutorial, you need a Bluemix account. Bluemix accounts provide access to everything you need to develop, track, plan, and deploy apps. You can sign up for a free 30-day trial.

# Create an app in Bluemix

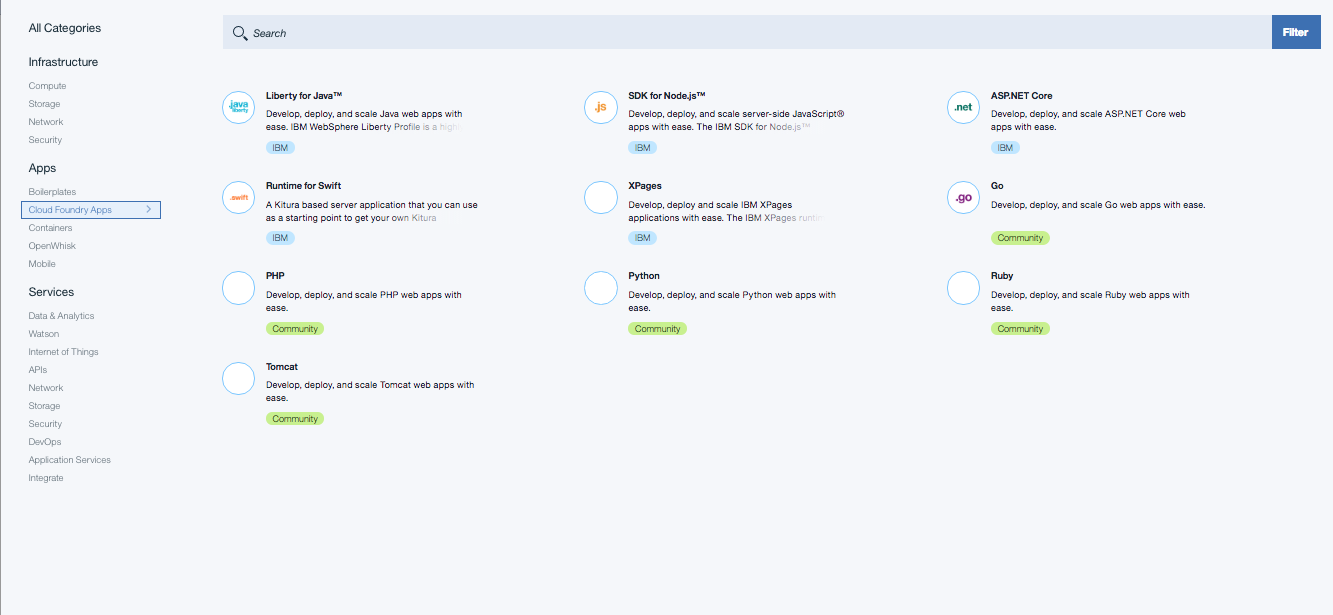
The Dashboard shows an overview of the active Bluemix space for your organization. By default, the space is dev and the organization is the project creator's user name. For example, if sara@example.com signs in to Bluemix for the first time, the active space is dev and the organization is sara@example.com.

If you create other organizations or spaces in Bluemix, be sure to use the default selections as you follow the tutorial.

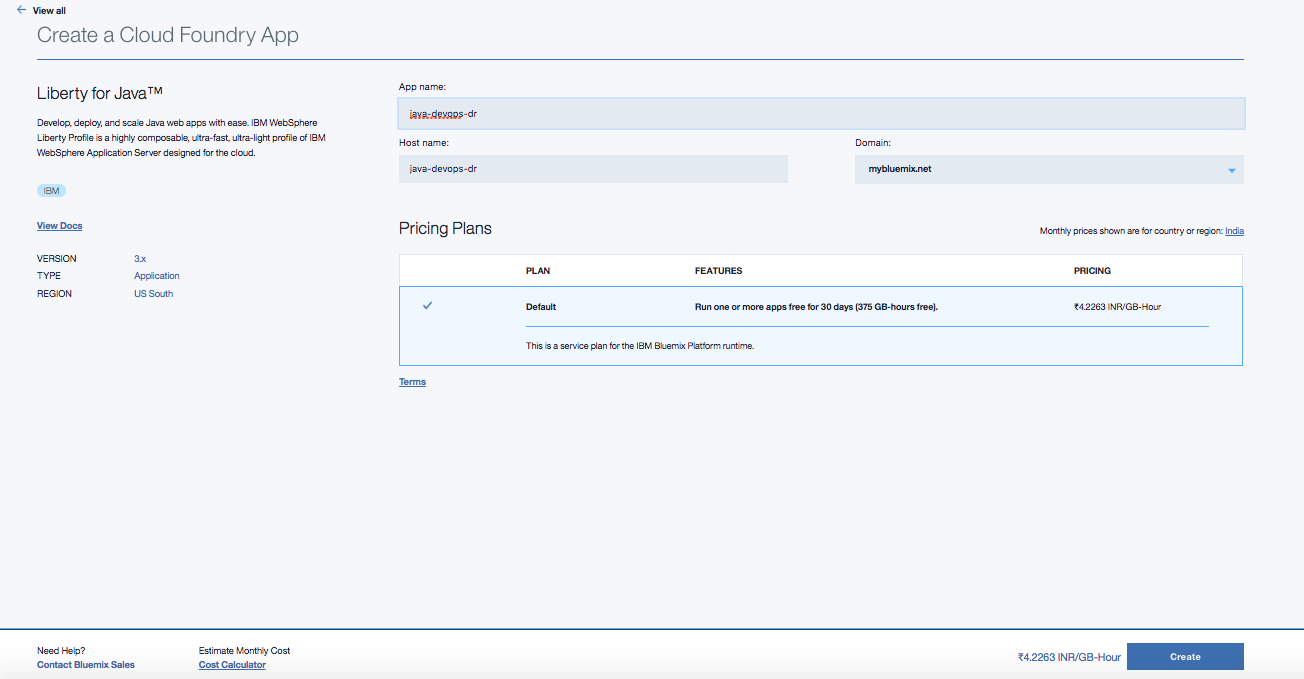
# Create the app

Click CREATE AN APP.

Under the cloud foundry apps, click Liberty for Java, and then click CONTINUE.

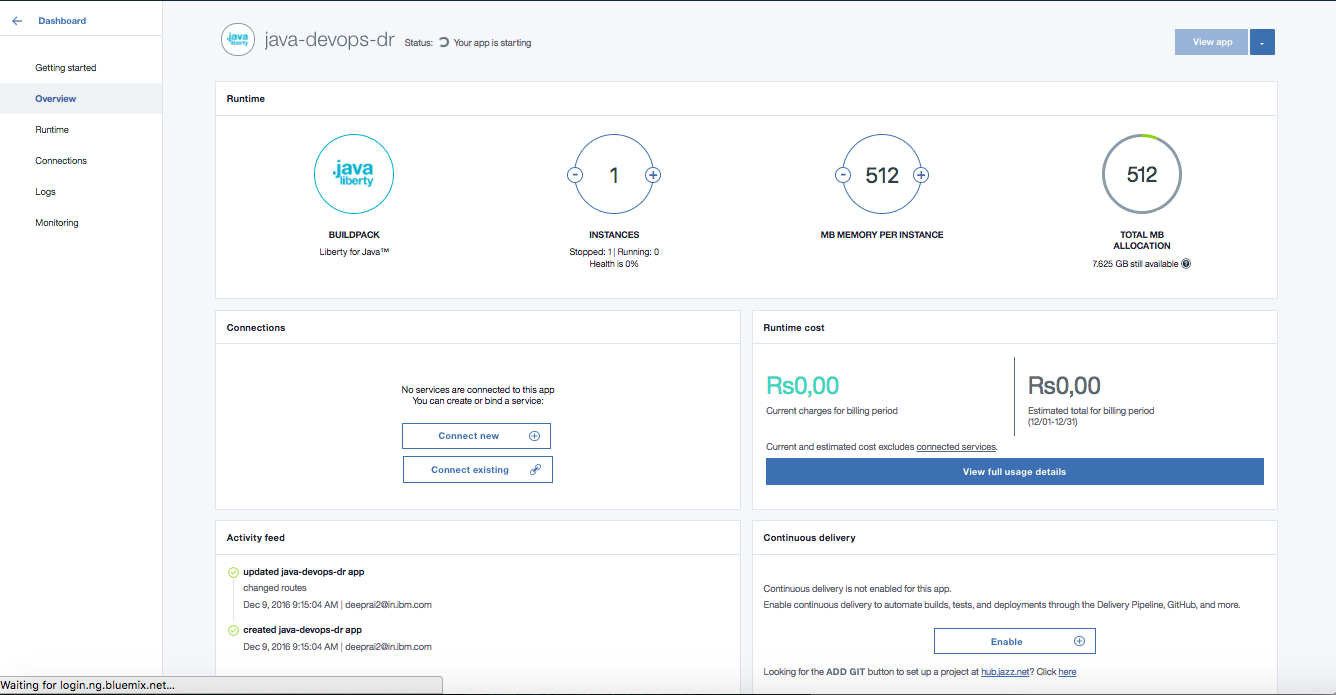


Type a unique name for your app, such as Liberty for Java Starter app tutorial <your initials>.

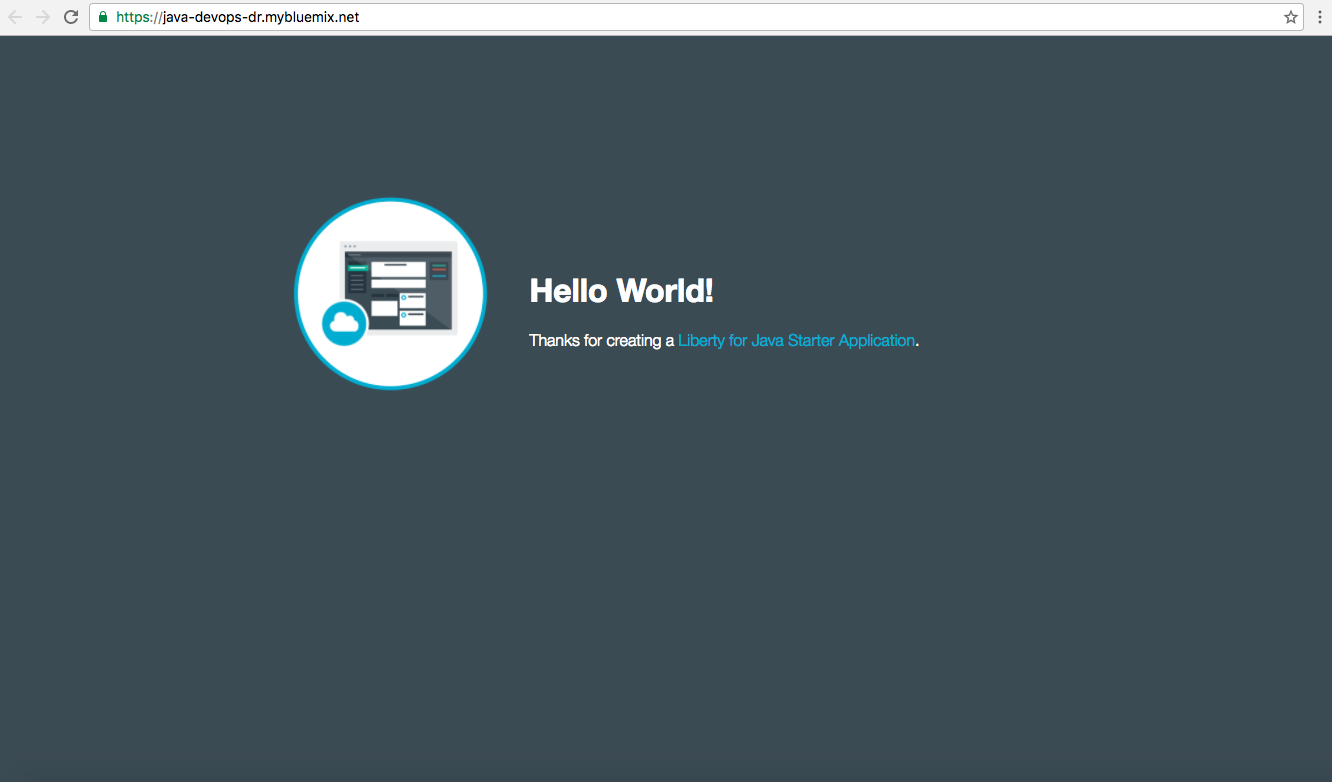


Click Finish. In a few moments, when the app is staged, your Java web server app is running on Bluemix.

Click Overview to view information about your app:



Click on View app on the app's Dashboard to access the running Starter app:

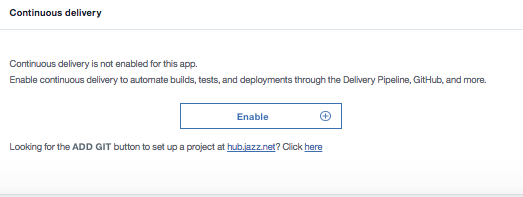


# Create a Git repository and enable auto-deploy

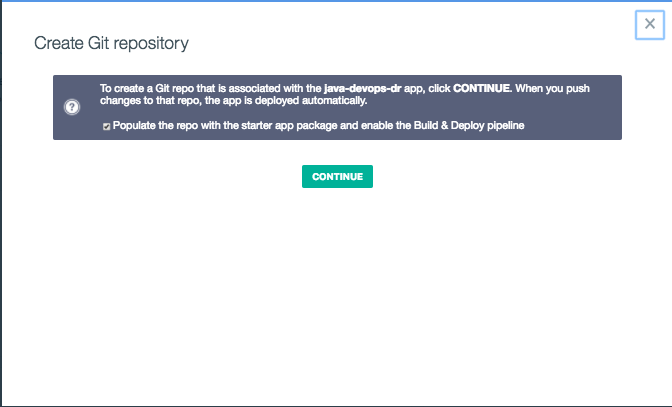
It's great that the app is running, but you want to make it more interesting.

You can work with your app code in many ways. For example, with auto-deploy, you can modify the code that runs in your app by pushing the changes to a Git repository. The mechanism that implements auto-deploy is provided by DevOps Services.

On your app’s Overview page, click Looking for the ADD GIT button to set up a project at hub.jazz.net? Click here. You might be prompted to enter your password because DevOps Services must verify that it can act on your behalf.



Select the Populate the repository with the starter application package and enable Delivery Pipeline (Build & Deploy) check box.



Click CONTINUE, and then click CLOSE. A message states that you created a Git repository and loaded it with the app starter code.

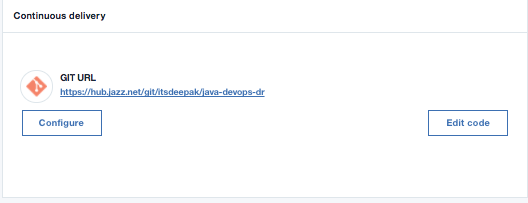
Optional

Add the Delivery Pipeline service to your space

* On the app’s Overview page, under Connections , click on CONNECT NEW
* Select the DevOps category, and then click Delivery Pipeline.
* Select the space that your app is in. Use the default settings for the selected plan.
* Click CREATE. The Delivery Pipeline service is added to your Bluemix space. You can view a list of the apps that include the service.
* Click Back to Dashboard and click the app tile for the app that you created.

# Start using DevOps Services

Click EDIT CODE.



Your project opens in the DevOps Services Web IDE (integrated development environment). When you clicked ADD GIT in Bluemix, a new Git repository was populated with sample code. When you clicked EDIT CODE, the sample code was loaded into this project workspace.

# Push changes to the Git repository by using the command line

You now have a Git repository with sample code. You can use the command-line Git support to push files from your computer to the repository.

If Git isn't installed, [see Setting up local clients to work with Git source control](https://hub.jazz.net/docs/gitclient/index.md). To learn more about Git, [see the Git documentation](http://git-scm.com/doc).

* Clone from the project repository URL, which you can find on the Bluemix app's Overview page or the project's Overview page in DevOps Services:

$ git clone https://hub.jazz.net/git/jazzhubdemouser/Liberty.for.Java.Starter.app.tutorial Cloning into 'Liberty.for.Java.Starter.app.tutorial'... Username for 'https://hub.jazz.net': <username> Password for 'https://<username>@hub.jazz.net': <password> remote: Counting objects: 49, done remote: Finding sources: 100% (49/49) remote: Total 49 (delta 0), reused 49 (delta 0) Unpacking objects: 100% (49/49), done. Checking connectivity... done.

* Change to the WebContent folder in the app's root directory and open the index.html file.
* Change some of the text in the body of the HTML.
* Change to your project directory and git stage, git commit, and git push:

$ cd /project\_directory $ git stage Webcontent/index.html $ git commit -m "Sample app headline change" [master 2336018] Sample app headline change 1 file changed, 1 insertion(+), 1 deletion(-) $ git push Username for 'https://hub.jazz.net': <username> Password for 'https://<username>@hub.jazz.net': Counting objects: 8, done. Delta compression using up to 4 threads. Compressing objects: 100% (4/4), done. Writing objects: 100% (4/4), 382 bytes | 0 bytes/s, done. Total 4 (delta 3), reused 0 (delta 0) remote: Resolving deltas: 100% (3/3) remote: Processing changes: refs: 1, done To https://hub.jazz.net/git/jazzhubdemouser/Liberty.for.Java.Starter.app.tutorial ab4aaad..2336018 master -> master

* In DevOps Services, click **BUILD & DEPLOY**. You can see that a new build was requested when changes were pushed to the repository. The build is automatically triggered by the changes that you delivered to the project repository. When the build is finished, it is deployed to Bluemix. When the instance is running, you can view your updated app on the web.

You can continue to modify the example and push your changes to the Git repository as often as needed. As the scope of the work grows and more people are added to the project, everyone can push changes to the repository. Standard repository operations and build and deploy are all that is required to ensure that the right bits are always running.

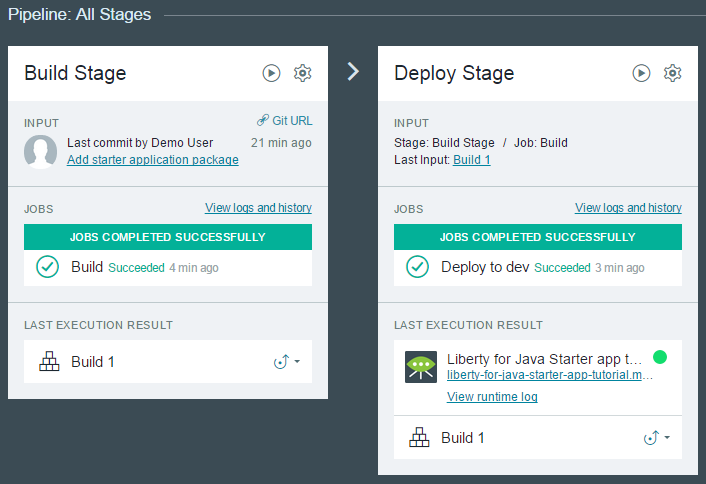
If you want to configure desktop clients to work with your Git repository, [see Setting up local clients to work with Git source control](https://hub.jazz.net/docs/gitclient.index.md).

# Explore builds and deployments

On the Build & Deploy Pipeline page, run your build by clicking the **Run Stage** icon un Stage icon on the Build Stage tile.

The build stage begins running. The deploy stage is configured to auto-deploy your app after the completion of a successful build stage.

When the app is running, you can view it by clicking the URL on the Deploy Stage tile. To see the app's Overview page on Bluemix, click the last execution result that shows the app's name.

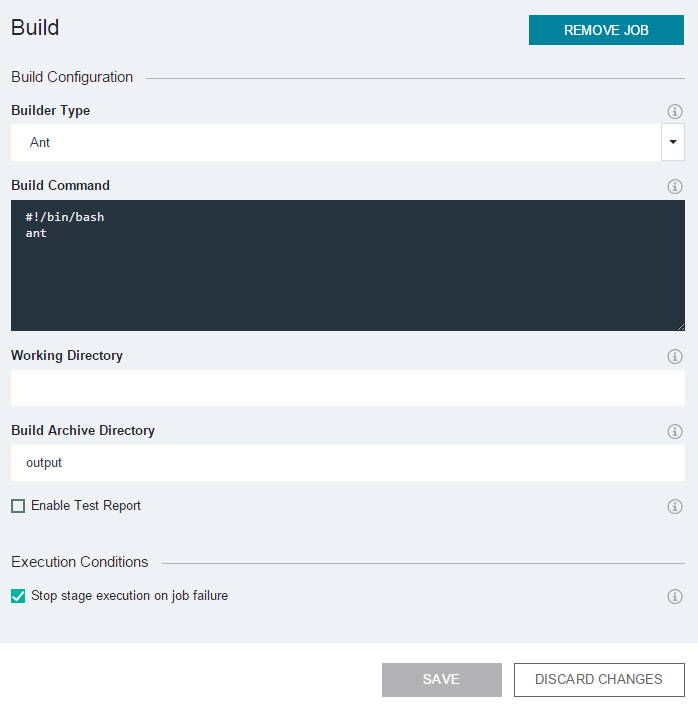


Your sample project is configured to automatically build and deploy the Liberty for Java sample app. You can explore the preconfigured stages or change the configurations.

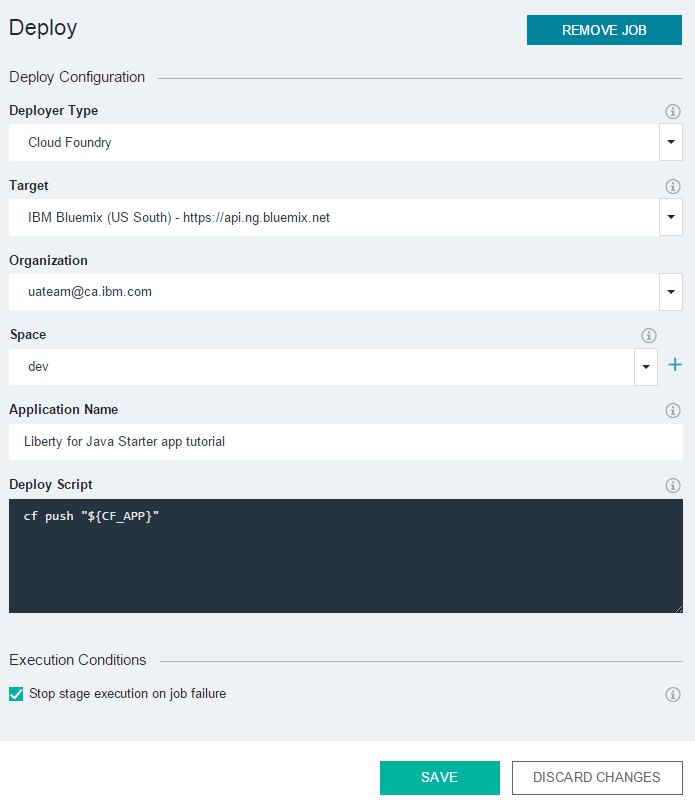
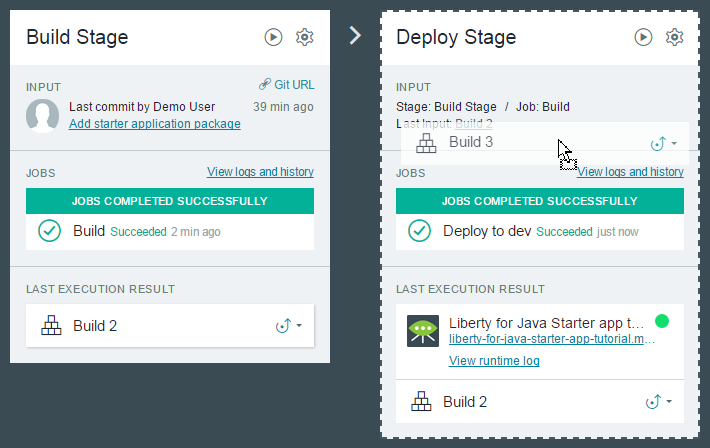
# Explore the preconfigured stages

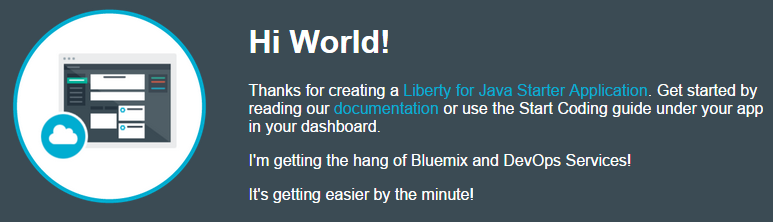
The Delivery Pipeline configuration for your app has two stages: a build stage and a deploy stage. These stages form a pipeline. The build stage runs a build job on the included build.xml. Then, the deploy stage runs a deploy job to deploy your code to Bluemix.

## Explore the build stage

* In the upper-right corner, click **BUILD & DEPLOY**.
* On the Build Stage tile, click the **Stage Configuration** icon tage Configuration icon and click **Configure stage**.
* Click the **INPUT** tab and review the settings:
* The input for the build is the master branch of the Git repository.
* The Build Stage runs automatically every time a change is pushed to the repository.
* Click the **JOBS** tab and review the settings:
* The builder type is **Ant**. The Ant command finds the default build script file, build.xml, which is provided in the sample at the root directory.
* Because the build.xml file is in the root of the project, you do not have to specify a working directory.
* The build output will be saved in an output directory, as specified in the **Build Archive Directory** field. The files that are needed for deployment are copied into the build archive directory.
* If the build is not completed, the stage stops running and any later jobs do not run.
* Because you did not make changes, click **DISCARD CHANGES** to return to the Build & Deploy Pipeline page.
* At the top of the Build Stage tile, click the **Run Stage** icon un Stage icon. After a moment, you have a successful build.

## Explore the deploy stage

* On the Deploy Stage tile, click the **Stage Configuration** icon "Stage
* and click **Configure stage**.
* Click the **INPUT** tab and review the settings:
* The input for the build is the output from the build stage.
* The deploy stage runs automatically every time the build stage runs successfully.
* Click the **JOBS** tab and review the settings:
* The app is set to deploy to your Bluemix organization and space.
* Bluemix is based on Cloud Foundry, so the deployment script uses the Cloud Foundry command-line interface command cf push to deploy your app. To learn more about how to configure deployment scripts when you use Cloud Foundry, [see the Cloud Foundry documentation](http://docs.cloudfoundry.org/devguide/installcf/whats-new-v6.html#push).
* Because you did not make changes, click **DISCARD CHANGES** to return to the Build & Deploy Pipeline page.
* Because you clicked the **Automatically execute jobs when the previous stage completes successfully** option when you set up the deploy stage, builds of your app are deployed automatically.
* **Tip**: In the future, if you do not click that option, you can start a deployment by dragging a build to a stage that has a deploy job. For example, you might drag the build from the LAST EXECUTION RESULT section of the build stage to the deploy stage, as shown here.
* 
* You can also start a deployment by clicking the **Run Stage** icon un Stage icon on the deploy stage. In a few moments, your app is deployed to Bluemix.
* To view your app on the web, on the Deploy Stage tile, click the URL in the LAST EXECUTION RESULT section. If the app is already open in a browser window, refresh the page. To view your app’s Bluemix Dashboard, click the app's name above the URL.



# Summary

In this tutorial, you learned to create and run a Java web app by using Bluemix.