IBM Bluemix DevOps Services V2 (BETA)

Note: Before you use this information and the product it supports, read the "Notices" section.

This edition applies to the IBM Bluemix DevOps Services V2 (beta) and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

Developing apps by using a toolchain (BETA)

Last updated: 16 December 2015

A *toolchain* is a set of complementary tools that create a flexible solution when you use them together. Integrations across a toolchain ease the workflows between the tools so that they work in synergy.

You can configure your app for continuous delivery by adding a toolchain to it. The toolchain includes a GitHub Enterprise repository (repo) that is populated with app starter code and a preconfigured delivery pipeline. The pipeline builds any changes to the GitHub Enterprise repo and then deploys the app to IBM® Bluemix®.

GitHub Enterprise is the on-premises version of GitHub. In addition to the features of GitHub, GitHub Enterprise provides security, hosting, and administrative options. It also includes 24x7 support for urgent issues. To learn more, see the GitHub Enterprise website.

Before you begin

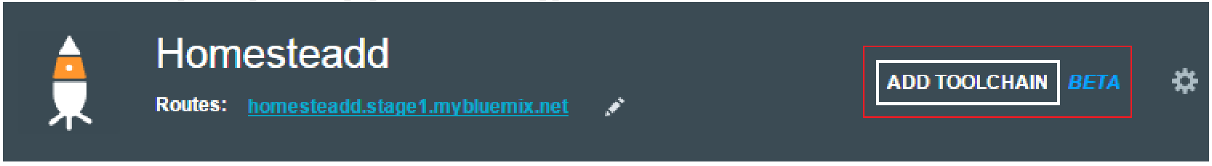
You need a Bluemix account. Bluemix accounts provide access to everything you need to develop, track, plan, and deploy apps.

Creating an app

1. Log in to Bluemix. The Dashboard opens and shows an overview of the active Bluemix space for your organization (org). By default, the space is dev and the org is your user name.
2. Click **CREATE APP**.
3. Click **WEB** and follow the guided experience to choose a starter, specify a name, and select how you want to code.
4. In the left navigation bar, click **Overview** to open the Overview page for your app in the Dashboard.

Setting up a toolchain

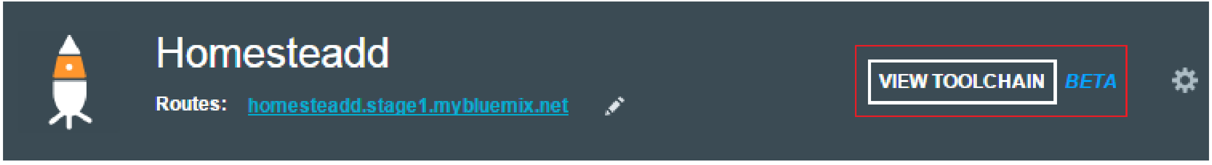
1. In the upper-right corner of the app's Overview page, click **ADD TOOLCHAIN**. Your app is configured for continuous delivery from a new GitHub Enterprise repo that is populated with the app starter code.



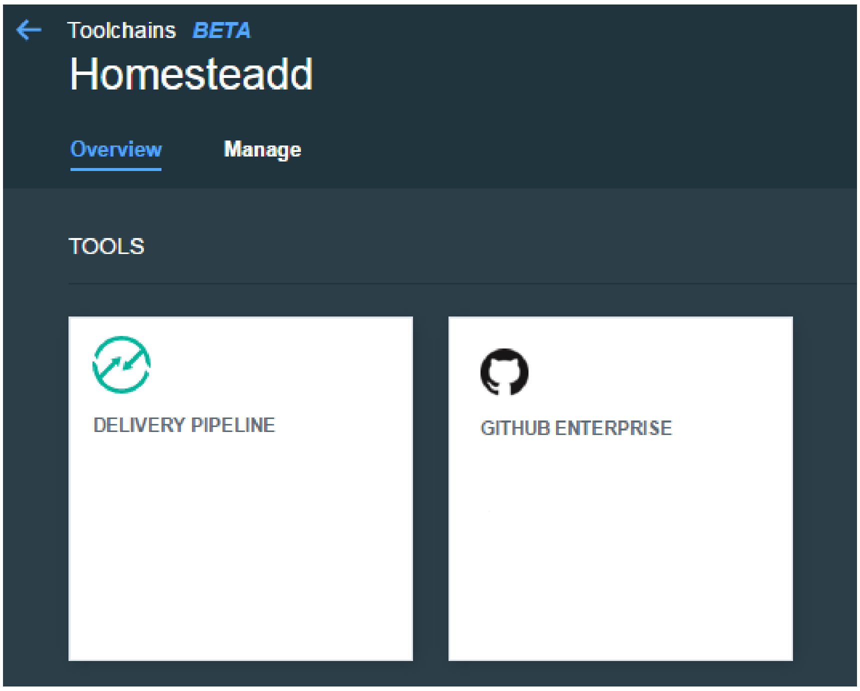
2. Click **CONTINUE** to create the toolchain. The toolchain is associated with your app. Each app has its own instance of a toolchain. When you push changes to the toolchain's GitHub Enterprise repo, the delivery pipeline automatically builds and deploys the app.

Viewing a toolchain

1. In the upper-right corner of the app's Overview page, click **VIEW TOOLCHAIN** to open the toolchain's Overview page.



The toolchain's Overview page provides a visual representation of the toolchain for your app.



2. Click the GitHub Enterprise tile to access the repo that is associated with your toolchain. You can view the activity that is associated with your app or push changes to the repo.

3. Click the Delivery Pipeline tile to access the pipeline that is associated with your toolchain. You can view the build and deployment activity or update the configuration for your app.

Deleting a toolchain

1. Click **VIEW TOOLCHAIN** and then click **Manage**.
2. Click **Delete Toolchain** and review the tools that you are deleting.
3. Confirm the deletion by typing the name of the toolchain and clicking **DELETE**.

Adding a user

You have three authentication options for your Bluemix dedicated environment: LDAP, SAML, or Web ID.

Important: For this beta, Web ID authentication requires additional user management on GitHub Enterprise.

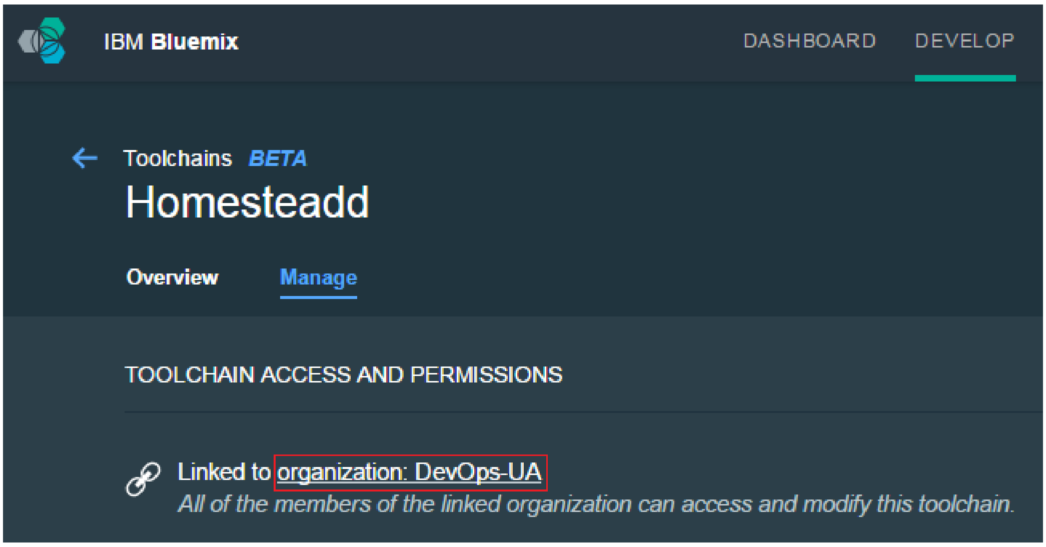
If you use LDAP or SAML authentication in your Bluemix dedicated environment, when you add users to your Bluemix org and spaces, the users can log in to GitHub Enterprise by using their Bluemix ID and password, and accounts are created for them. When you add users to your Bluemix org and spaces, they are not automatically added to the GitHub Enterprise repo. Someone who has admin privileges for the repo must add them.

If you use Web ID authentication, when you add users to your Bluemix org and spaces, a GitHub Enterprise site administrator must set up a GitHub Enterprise account for those users. Alternatively, new users can create a toolchain, in which case a GitHub Enterprise account is created for them. However, if those users want to access repos that are associated with toolchains besides their own, they must be granted access to those repos.

To add a user:

1. From your app's toolchain, click **Manage**.

2. Click the link to your organization.



Homesteadd

TOOLCHAIN ACCESS AND PERMISSIONS

Linked to organization: DevOps-UA

All of the members of the linked organization can access and modify this toolchain.

3. On the Manage Organizations page, click **ADD USER**, type the user's name, and click **ADD**.

   Note: The user name might be an email address.

4. If you want to give advanced permissions to the user, select one or more of the **MANAGER**, **BILLING MANAGER**, or **AUDITOR** check boxes.

5. Click **SAVE**.

Tutorial: Continuously integrate and deliver by using a toolchain (BETA)

Last updated: 16 December 2015

Time: 30 minutes

A *toolchain* is a set of complementary tools that create a flexible solution when you use them together. Integrations across a toolchain ease the workflows between the tools so that they work in synergy. You can use a toolchain to support the continuous integration and delivery of your apps.

In this tutorial, you create an app in IBM® Bluemix® by using the SDK for Node.js starter pack. You then add a toolchain to your app to create a GitHub Enterprise repository (repo) and save your app source in the repo. The toolchain is associated with both the GitHub Enterprise repo and your app. Next, you make a few changes and push them to the GitHub Enterprise repo. When you push changes to your GitHub Enterprise repo, a delivery pipeline, which can build and deploy the app to Bluemix, is generated.

GitHub Enterprise is the on-premises version of GitHub. In addition to the features of GitHub, GitHub Enterprise provides security, hosting, and administrative options. It also includes 24x7 support for urgent issues. To learn more, see the GitHub Enterprise website.

Before you begin

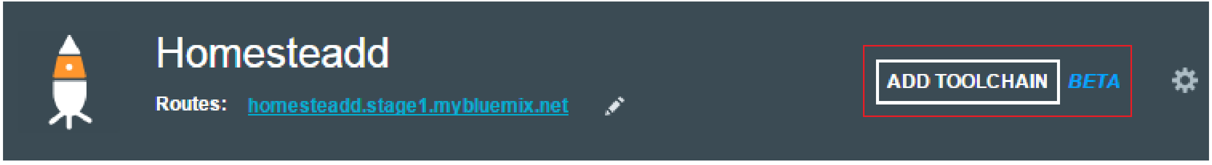
To do this tutorial, you need a Bluemix account. Bluemix accounts provide access to everything you need to develop, track, plan, and deploy apps.

Create an app

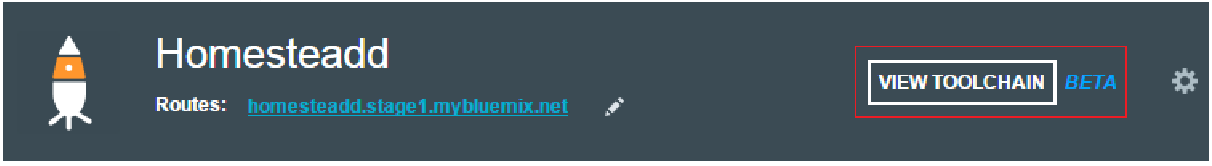
1. Log in to Bluemix. The Dashboard opens and shows an overview of the active Bluemix space for your organization (org). By default, the space is dev and the org is your user name.
2. Click **CATALOG**.
3. Select the **SDK for Node.js** runtime and review the documentation and terms.
4. Name your app and type a value in the **Host** field. That value is used in a unique URL that you can use to access your app. Click **CREATE**. After a moment, the app starts and the Start Coding page opens.
5. In the left navigation bar, click **Overview** to open the Overview page for your app in the Dashboard.

Add a toolchain to your app

1. In the upper-right corner of the app's Overview page, click **ADD TOOLCHAIN**. Your app is configured for continuous delivery from a new GitHub Enterprise repo that is populated with the app starter code.



2. Click **CONTINUE** to create the toolchain. The toolchain is associated with your app. Each app has its own instance of a toolchain. When you push changes to the toolchain's GitHub Enterprise repo, the delivery pipeline automatically builds and deploys the app.
3. To view and begin working with your toolchain, click **VIEW TOOLCHAIN** from the Overview page.



Add a stage to your pipeline

1. If you haven't opened the toolchain yet, click **VIEW TOOLCHAIN**. On the Overview page, click the **DELIVERY PIPELINE** tile to view the build and deployment activity for your app.

   As a result of the toolchain creation process, your toolchain has a functional pipeline in place already. The pipeline has two stages: one where your app is built, and another where it is deployed. You need to add a third stage to deploy a test instance of your app.

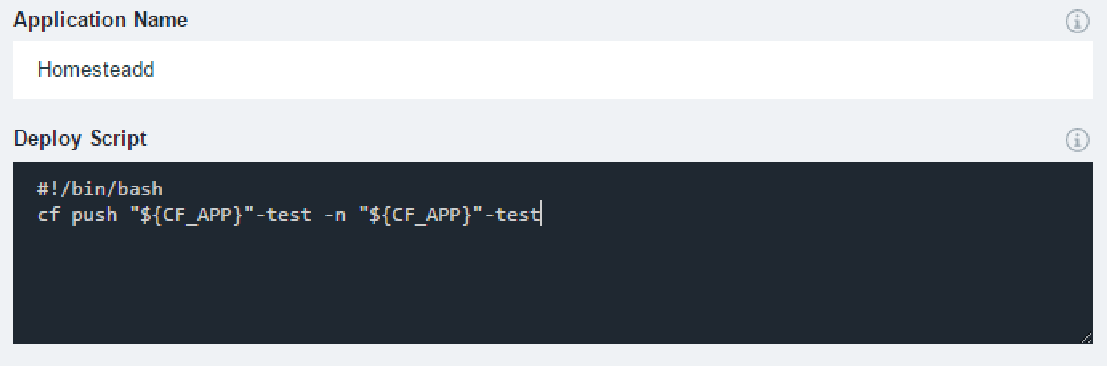
2. Click **ADD STAGE**.

3. Click the stage name, **MyStage**, and change it to `Deploy Stage - Test`.

4. Click the **JOBS** tab.

5. Click **ADD JOB**, and select the **Deploy** job type.

6. In the **Deploy Script** field, change the `cf push` command to `cf push "${CF_APP}"-test -n "${CF_APP}"-test`.

   Application Name

   Homesteadd

   Deploy Script

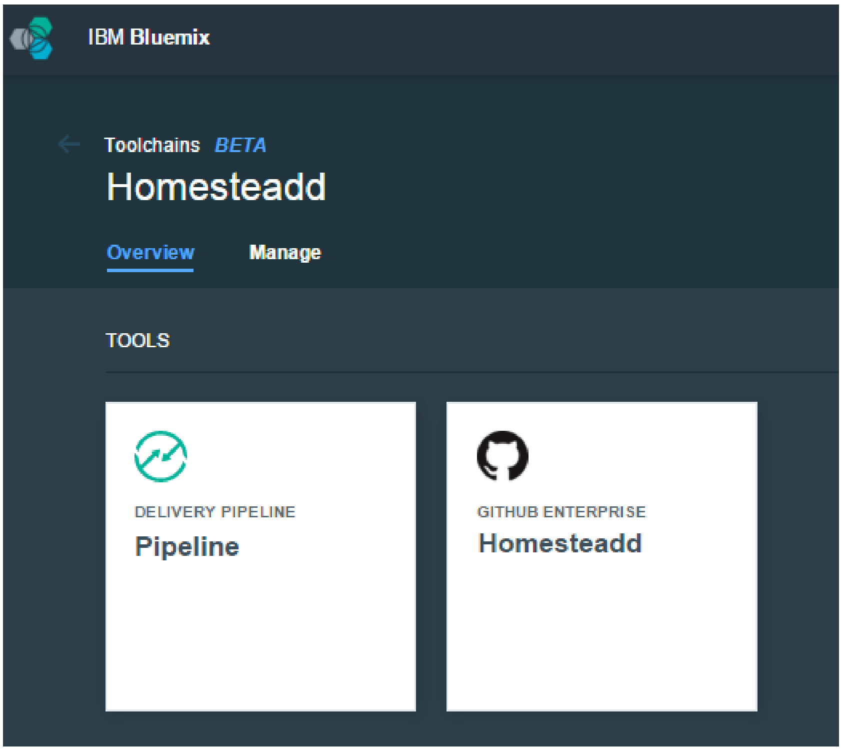
   ```
   #!/bin/bash
   cf push "${CF_APP}"-test -n "${CF_APP}"-test
   ```

7. Click **SAVE**.

8. Click the **Deploy Stage - Test** title on the new stage, and drag it between the first two stages.

Modify your app's code

1. On the toolchain's Overview page, click the **GITHUB ENTERPRISE** tile to go to your repo.



IBM Bluemix

Toolchains BETA

Homesteadd

Overview Manage

TOOLS

DELIVERY PIPELINE

Pipeline

GITHUB ENTERPRISE

Homesteadd

2. In the GitHub Enterprise UI, navigate to the `index.html` file in the public folder and click **Edit this file**.
   Note: If you prefer to work from the command line, you can clone your repo and edit in your favorite editor.

3. Make a change to the heading or the text, add a comment, and click **Commit changes**.

4. Return to your toolchain's Overview page and click the **DELIVERY PIPELINE** tile to watch the stages run.

5. When the pipeline process is complete, go to your app's Overview page on the Dashboard, click the **Routes** link, and verify that your changes are visible in the running app.

Summary

You learned the basics of how to continuously integrate and deploy an app to Bluemix by using a toolchain.

You created an app and added a toolchain to it that included a GitHub Enterprise repo and a delivery pipeline. You added a stage to the pipeline, made a change to your app's code, and when you pushed the updated code, your changes were built and deployed automatically.

Next steps

Now that you know the basics, explore this information:

- DevOps Services Docs
- Bluemix Docs

Tutorial: Deliver continuously by using a pipeline (BETA)

Last updated: 16 December 2015

Time: 30 minutes

In this tutorial, you create an IBM® Bluemix® DevOps Services toolchain from a Bluemix starter app. That toolchain includes the Delivery Pipeline service (pipeline) and GitHub Enterprise. The pipeline automates the continuous deployment of your app. In a pipeline, sequences of stages retrieve input and run jobs, such as builds, tests, and deployments.

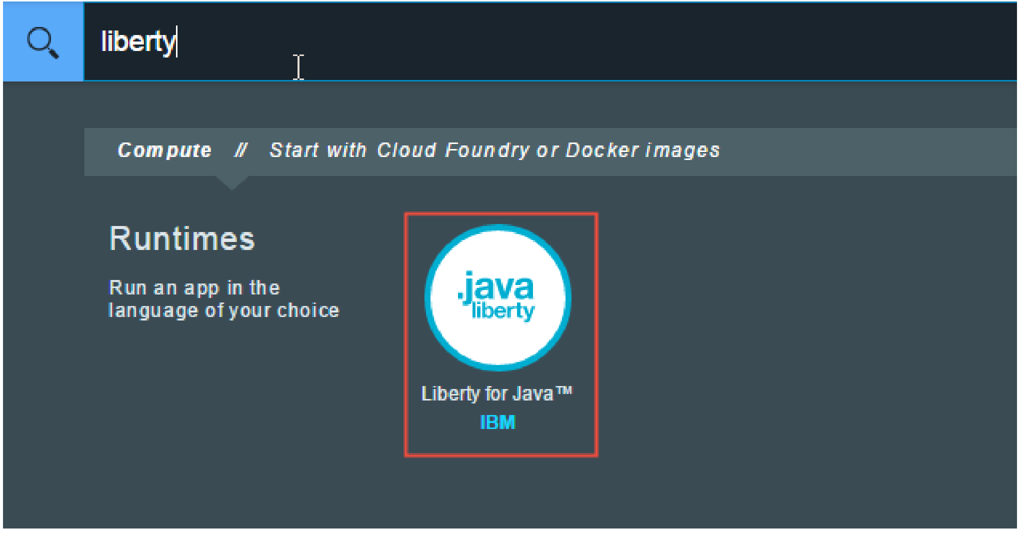
Over the course of this tutorial project, you set up a basic pipeline that you can use to build and deploy a Java™ app to test and production environments.

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.

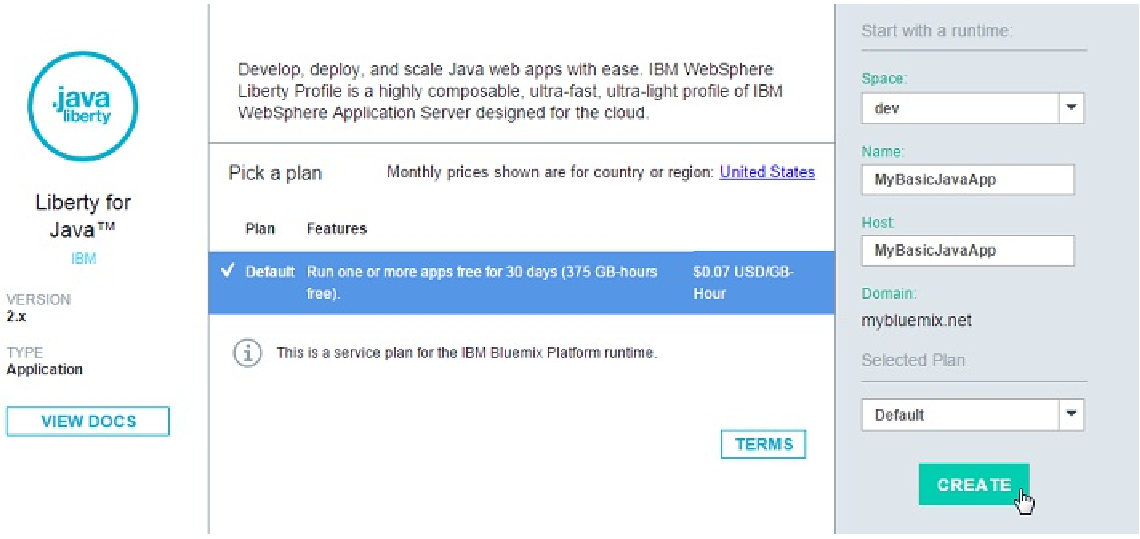
Create an app

1. Log in to Bluemix. The Dashboard opens and shows an overview of the active Bluemix space for your organization (org). By default, the space is dev and the org is your user name.
2. Click **CATALOG**.
3. Select the **Liberty for Java™** runtime and review the documentation and terms.



liberty
Compute // Start with Cloud Foundry or Docker images
Runtimes
Run an app in the language of your choice
.java liberty
Liberty for Java™
IBM

4. Name your app and type a value in the **Host** field. That value is used in a unique URL that you can use to access your app. Click **CREATE**. After a moment, the app starts and the Start Coding page opens.



Liberty for Java™

IBM

VERSION 2.x

TYPE Application

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.

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

Plan	Features	
✓ Default	Run one or more apps free for 30 days (375 GB-hours free).	$0.07 USD/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Start with a runtime:

Space: dev

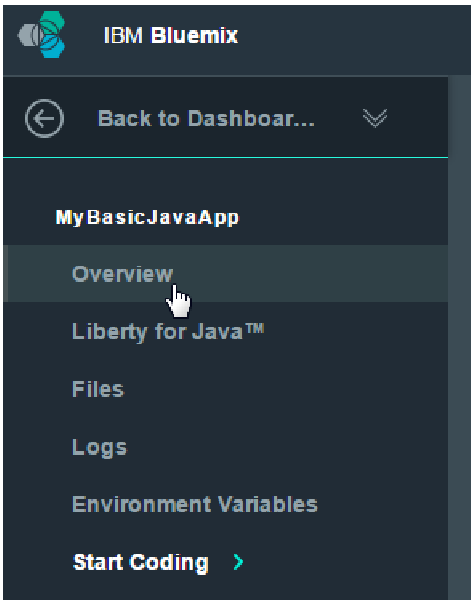
Name: MyBasicJavaApp

Host: MyBasicJavaApp

Domain: mybluemix.net

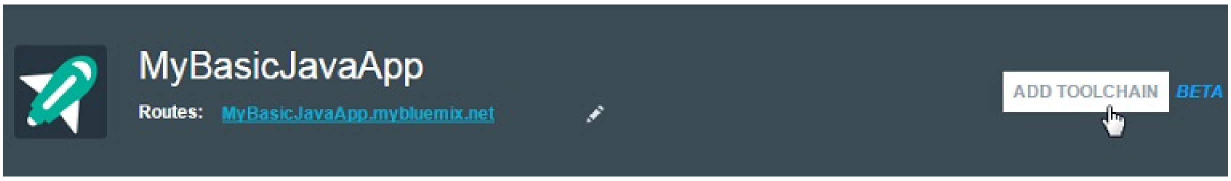
Selected Plan: Default

5. In the left navigation bar, click **Overview** to open the Overview page for your app in the Dashboard.



Add a toolchain to your app

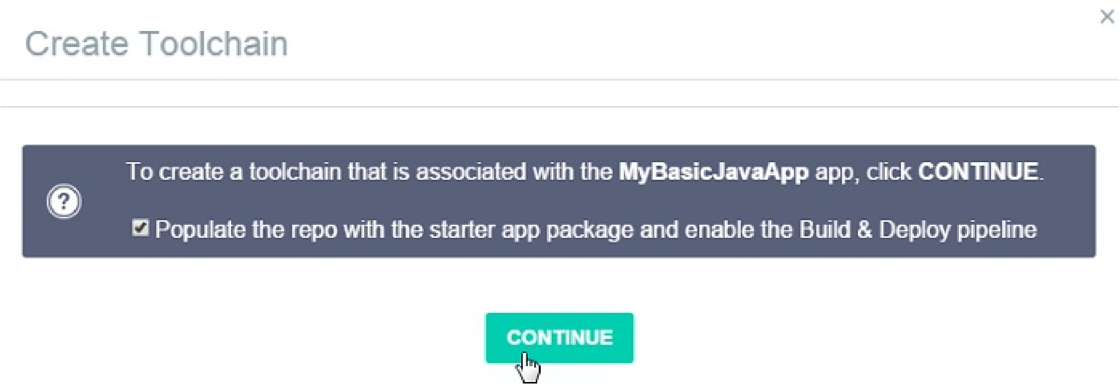
1. In the upper-right corner of the app's Overview page, click **ADD TOOLCHAIN**. A toolchain is created; it includes a delivery pipeline and a GitHub Enterprise repository that contains your app's source code.



MyBasicJavaApp

Routes: MyBasicJavaApp.mybluemix.net

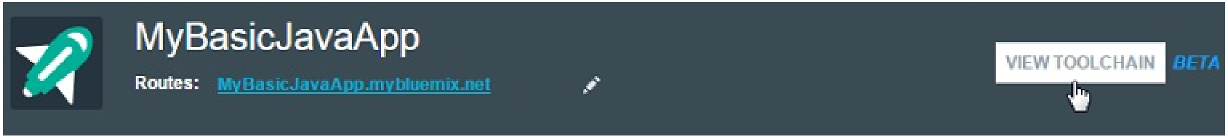
2. Click **CONTINUE** to create the toolchain. The toolchain is associated with your app. Each app has its own instance of a toolchain. When you push changes to the toolchain's GitHub Enterprise repo, the pipeline automatically builds and deploys the app.


Create Toolchain

To create a toolchain that is associated with the **MyBasicJavaApp** app, click **CONTINUE**.

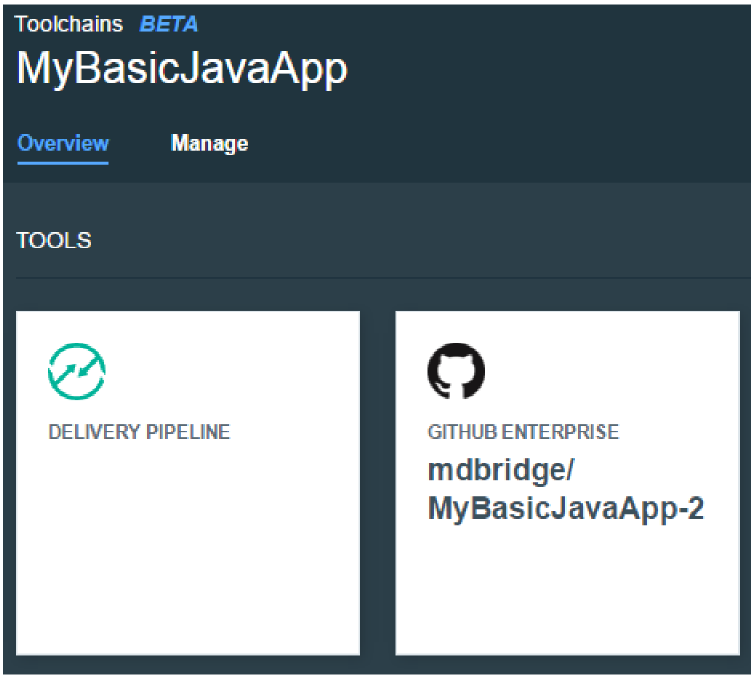
☑ Populate the repo with the starter app package and enable the Build & Deploy pipeline

3. After the toolchain is created, click **CLOSE**.
4. To view and begin working with your toolchain, click **VIEW TOOLCHAIN** from the Overview page.



Explore your pipeline

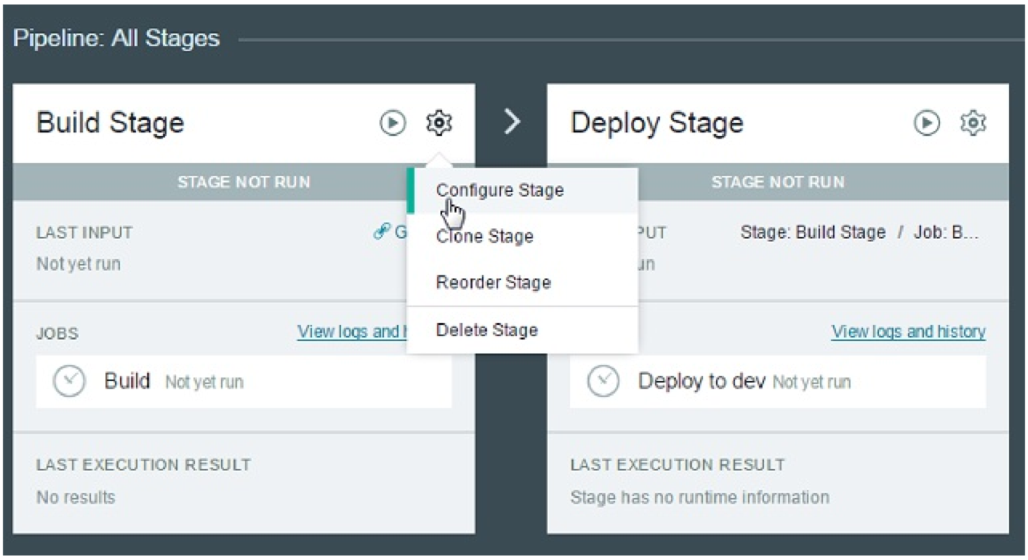
On the toolchain's Overview page, click the **DELIVERY PIPELINE** tile to view the pipeline for your app.



As a result of the toolchain creation process, your toolchain has a functional pipeline in place already. Pipelines consist of jobs and stages. Stages organize input and run jobs to build, deploy, and test your code. Jobs do the building, deploying, and testing.

Your pipeline has two stages: one that contains a job that builds your app, and another that contains a job that deploys your app. Explore their configurations to understand how they work and what they do.

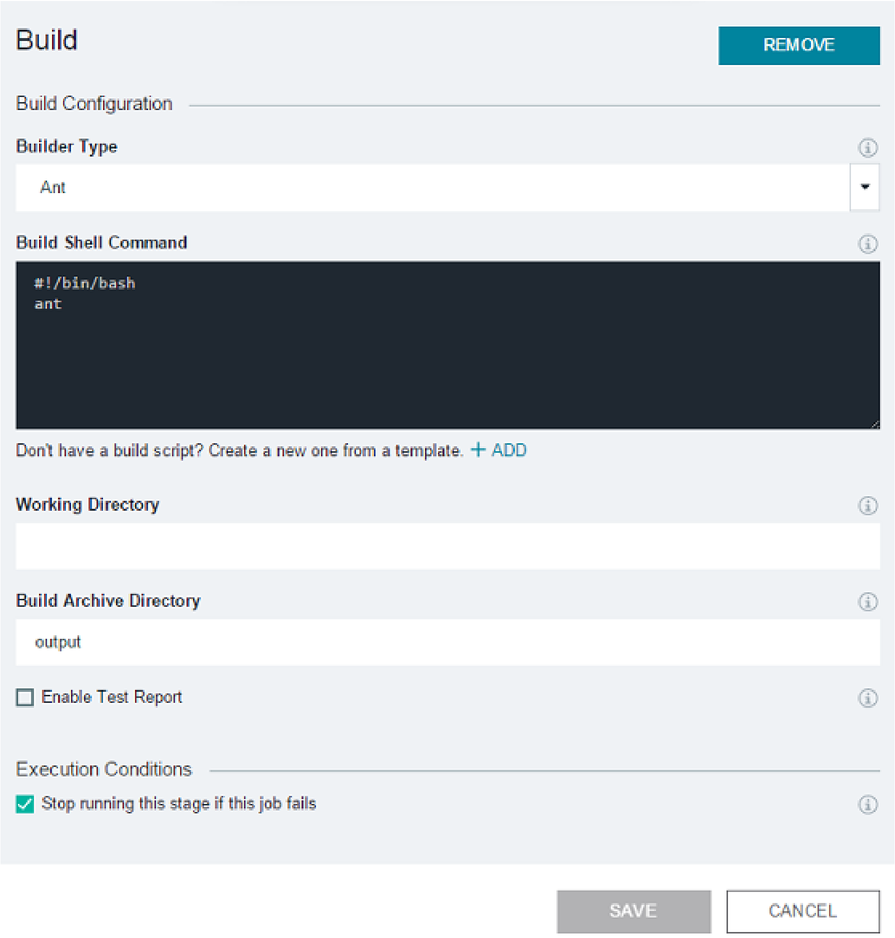
1. On the first stage, Build Stage, click the gear icon. Click **Configure Stage**.



a. Click the **INPUT** tab. Stages accept input from either source control repositories or build jobs in other stages. As you can see, this stage's input is from the master branch of your toolchain's GitHub Enterprise repository. By default, the pipeline runs in its entirety whenever a change is pushed to that branch.

b. Look at this stage's job: click the **JOBS** tab.

The stage contains one build job, which uses Ant to build the Java sample project. The results of the build are archived to the project's `/output` directory.


Build

Build Configuration

Builder Type: Ant

Build Shell Command

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

Don't have a build script? Create a new one from a template. + ADD

Working Directory:

Build Archive Directory: output

□ Enable Test Report

Execution Conditions

☑ Stop running this stage if this job fails

Jobs in a stage can't pass artifacts to each other; they're each given a clean container environment to run their stage's input in. However, you can define stage environment properties that can be used in all of a stage's jobs. For example, you might define a `TEST_URL` property that passes a single URL to deploy and test jobs in a single stage. If you click the **ENVIRONMENT PROPERTIES** tab, you'll see that this sample does not use any environment properties.

2. Next, look at the Deploy Stage:

a. Click **CANCEL**.

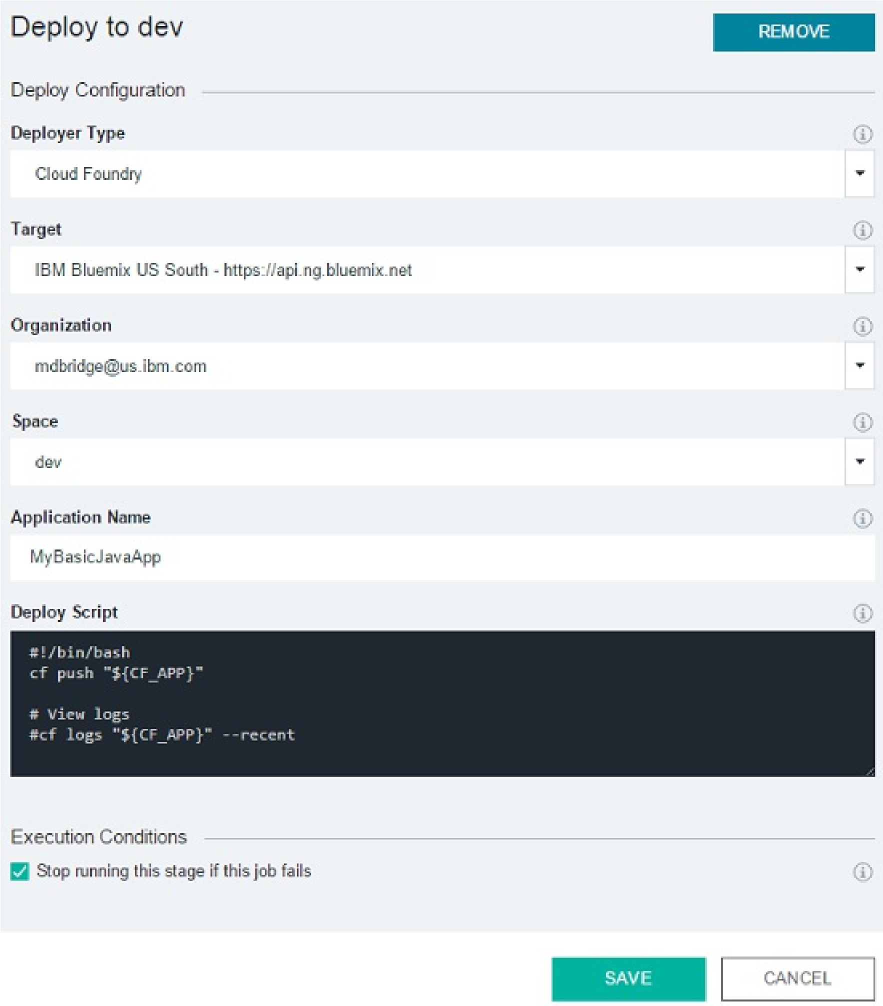
b. Click the gear icon on the second stage, Deploy Stage. Click **Configure Stage**.

c. Click the stage name, **Deploy Stage**, and change it to `Deploy Stage - Production`.

d. Click the **INPUT** tab.

This stage takes the build artifacts from the build job in the previous stage as input. Recall that the results of the build job are archived to the `/output` directory.

e. Click the **JOBS** tab. The stage contains one deploy job, which deploys your app to a Bluemix server that is controlled by your org.



Deploy to dev

Deploy Configuration

Deployer Type: Cloud Foundry

Target: IBM Bluemix US South - https://api.ng.bluemix.net

Organization: mdbridge@us.ibm.com

Space: dev

Application Name: MyBasicJavaApp

Deploy Script

```
#!/bin/bash
cf push "${CF_APP}"

# View logs
#cf logs "${CF_APP}" --recent
```

Execution Conditions

☑ Stop running this stage if this job fails

The `cf push ${CF_APP}` command in the **Deploy Script** field pushes an app with the name you specify in the **Application Name** field to the targeted Bluemix space. The Bluemix organization and space are selected with the Organization and Space boxes. The route that the app is deployed on is, in this case, determined by the manifest file in your GitHub Enterprise repository's root directory. For more details about manifest files, see the Bluemix Docs.

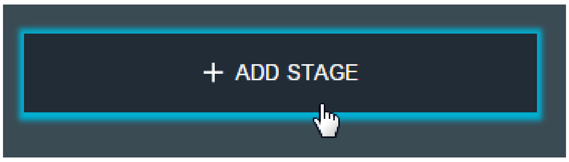
The default settings are sufficient for this pipeline, but you can view help for any option by clicking the help icon that is next to it.

3. Click **SAVE**.

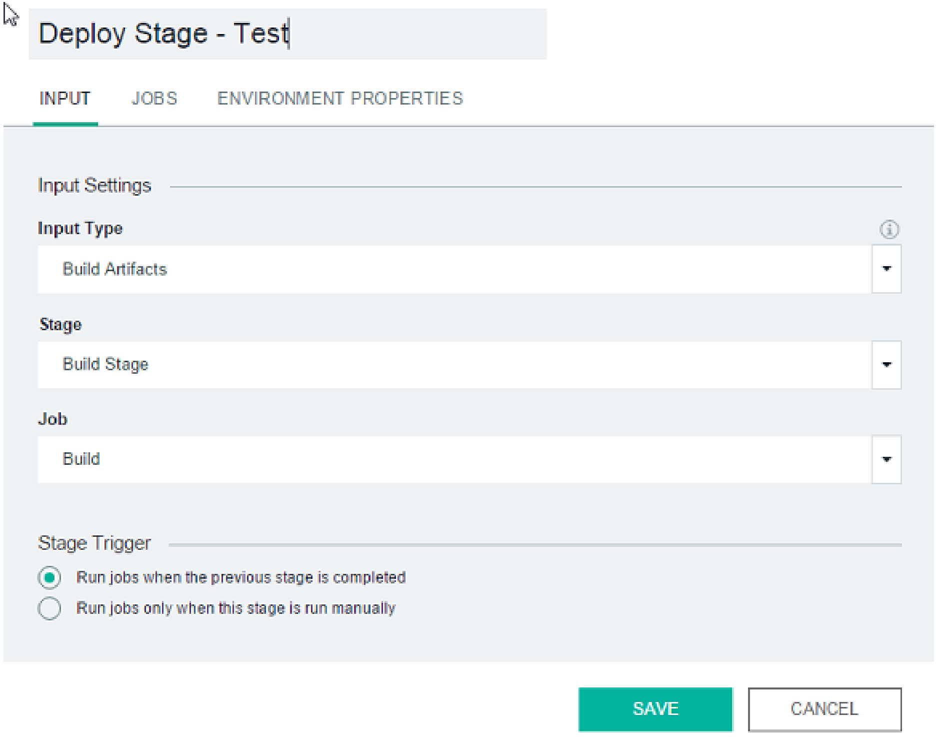
Add a stage to your pipeline

Next, add another stage to deploy a test instance of your app. The input is the same as Deploy Stage - Production, but the app's name is different. You'll also add a test job after your deploy job. You don't need to include any of your own tests; this exercise just gives you an idea of how the flow of control works in pipelines.

1. Click **ADD STAGE**.



2. Click the stage name, **MyStage**, and change it to `Deploy Stage - Test`.



Deploy Stage - Test

Input Settings

Input Type: Build Artifacts

Stage: Build Stage

Job: Build

Stage Trigger

● Run jobs when the previous stage is completed

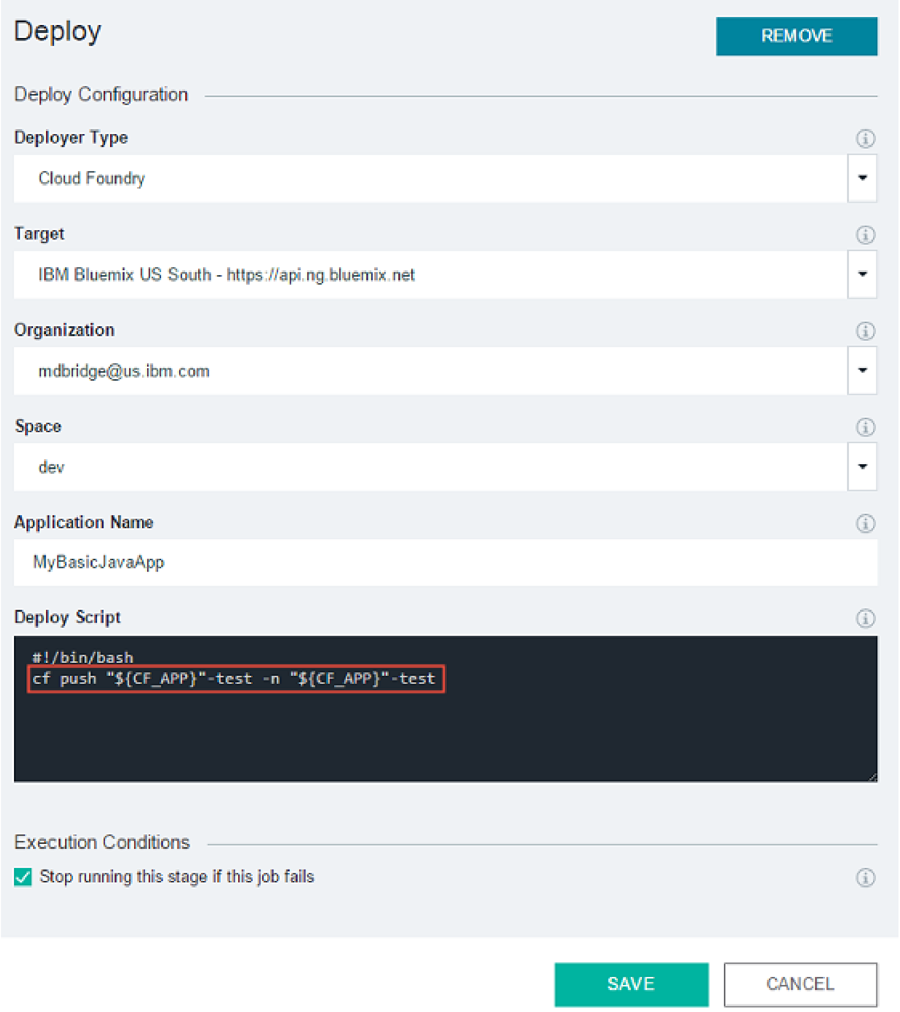
○ Run jobs only when this stage is run manually

3. Click the **JOBS** tab.

4. Click **ADD JOB**, and select the **Deploy** job type.

5. In the **Deploy Script** field, change the `cf push` command to `cf push "${CF_APP}"-test -n "${CF_APP}"-test`.

    - The `-n` flag allows you to set your app's route when you use the `cf push` command. Here, your app is pushed with its name followed by "-test." When the deploy job is completed, the app is available at a URL that includes its name followed by "-test."



Deploy

Deploy Configuration

Deployer Type: Cloud Foundry

Target: IBM Bluemix US South - https://api.ng.bluemix.net

Organization: mdbridge@us.ibm.com

Space: dev

Application Name: MyBasicJavaApp

Deploy Script:

```
#!/bin/bash
cf push "${CF_APP}"-test -n "${CF_APP}"-test
```

Execution Conditions

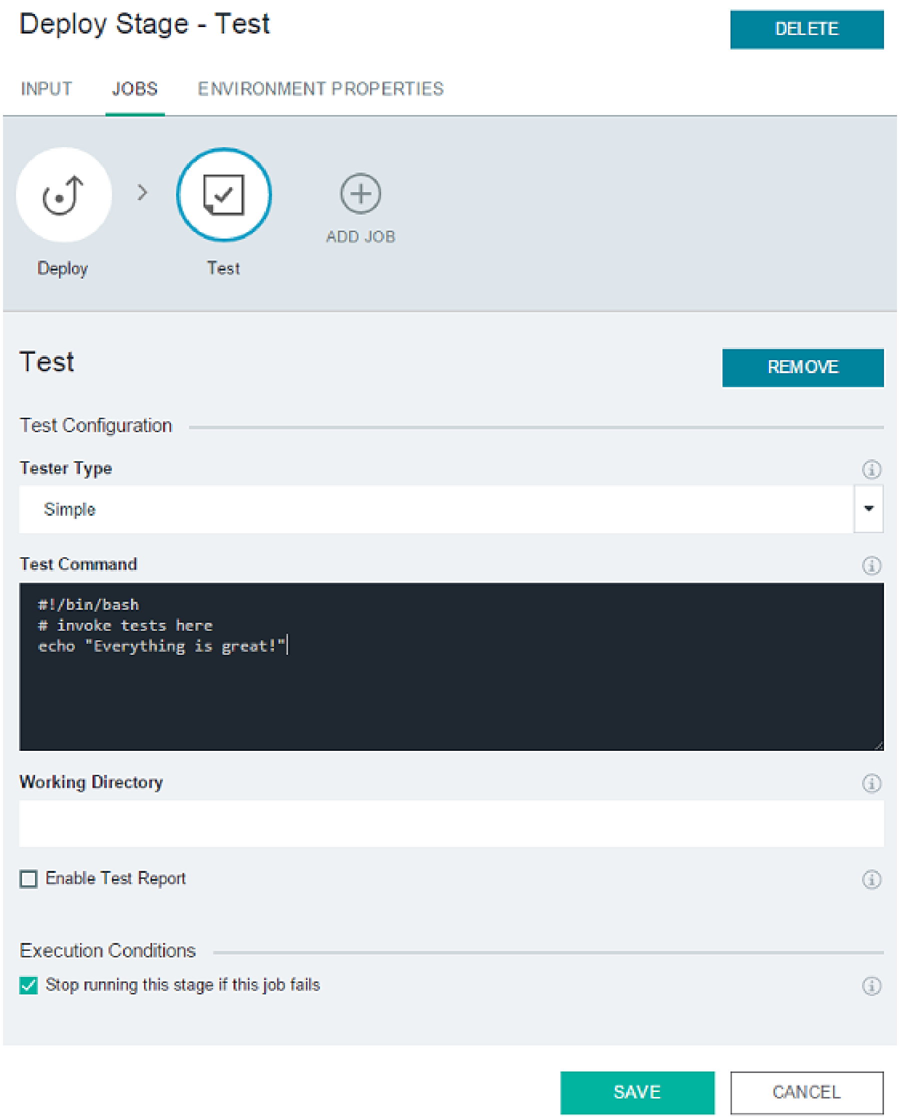
☑ Stop running this stage if this job fails

6. At the top of the screen, click **ADD JOB** again. For the job type, select **Test**.

6. At the top of the screen, click **ADD JOB** again. For the job type, select **Test**.

7. In the **Test Command** field, type `echo "Everything is great!"`.

   Tip: You can run more useful tests of your own design here, of course. You might write a test script in this field, or call test files that are in the stage input.



Tester Type: Simple

Test Command

```
#!/bin/bash
# invoke tests here
echo "Everything is great!"
```

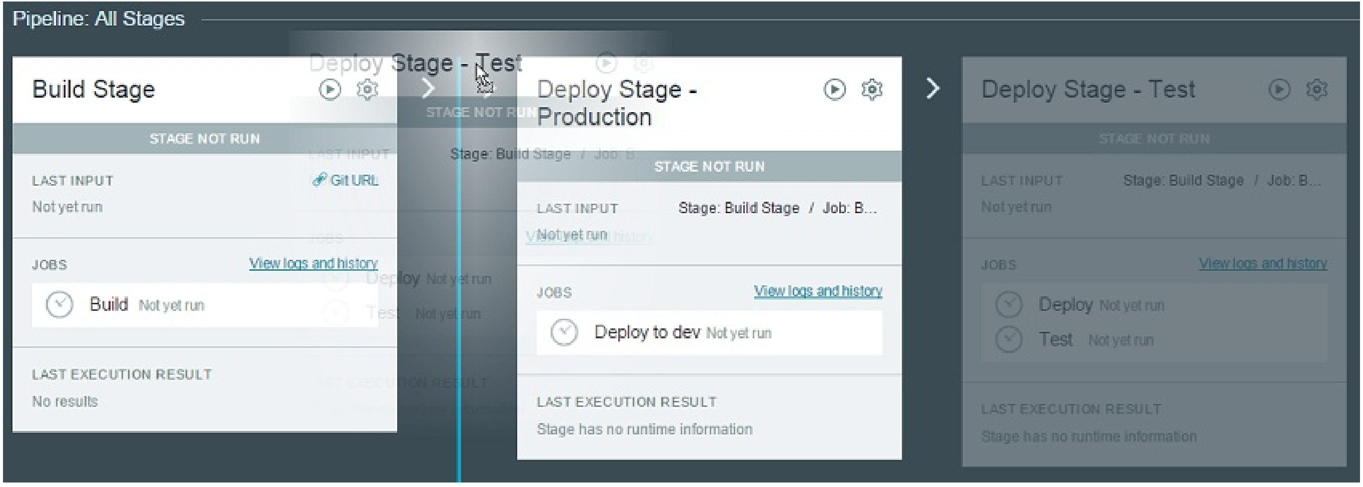
Working Directory

□ Enable Test Report

☑ Stop running this stage if this job fails

8. Click **SAVE**.

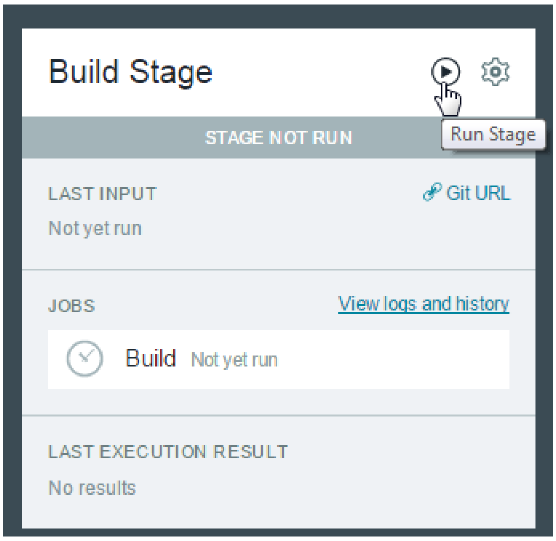
9. Click the **Deploy Stage - Test** title on the new stage, and drag it between the first two stages.



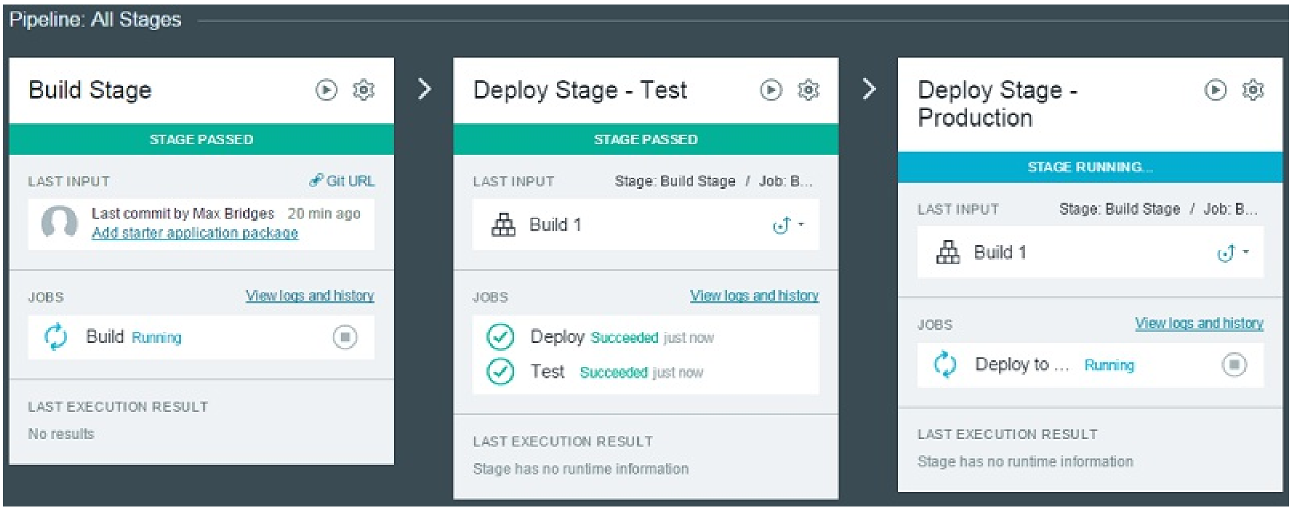
Stages and jobs run serially; they enable flow control for your work. Deploy Stage - Test is before Deploy Stage - Production. By default, if the test job in Deploy Stage - Test fails, the entire Deploy Stage - Test fails. This means that Deploy Stage - Production won't run. The middle test stage acts as a gatekeeper, preventing bad builds from getting to production.

Verify your pipeline

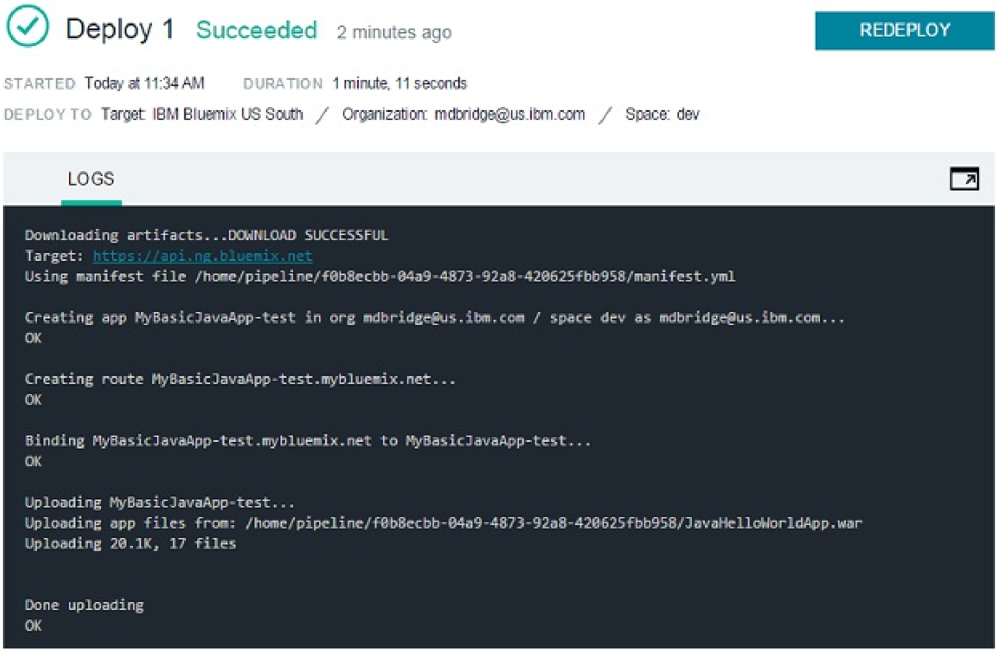
This part of the tutorial is the satisfying part, and there's only one step! At the top of the first stage, Build Stage, click the **Run Stage** icon. Usually, you don't need to do this manually. The pipeline will run whenever a change is pushed to the branch of the Git repository that it uses as input.



Watch your project as it is built in one stage, deployed and tested in a test stage, and then deployed to production in the final stage.



You can click any job to view its logs—even in real time.



STARTED Today at 11:34 AM DURATION 1 minute, 11 seconds

DEPLOY TO Target: IBM Bluemix US South / Organization: mdbridge@us.ibm.com / Space: dev

```
Downloading artifacts...DOWNLOAD SUCCESSFUL
Target: https://api.ng.bluemix.net
Using manifest file /home/pipeline/f0b8ecbb-04a9-4873-92a8-420625fbb958/manifest.yml

Creating app MyBasicJavaApp-test in org mdbridge@us.ibm.com / space dev as mdbridge@us.ibm.com...
OK

Creating route MyBasicJavaApp-test.mybluemix.net...
OK

Binding MyBasicJavaApp-test.mybluemix.net to MyBasicJavaApp-test...
OK

Uploading MyBasicJavaApp-test...
Uploading app files from: /home/pipeline/f0b8ecbb-04a9-4873-92a8-420625fbb958/JavaHelloWorldApp.war
Uploading 20.1K, 17 files


Done uploading
OK
```

Summary

In this tutorial, you added a toolchain by using a Bluemix starter, modified its auto-generated pipeline, and deployed it as test and production instances. You learned about flow control in pipelines, and how to use jobs to do what you want.

Notices

For the IBM Bluemix DevOps Services V2 beta terms of use, see IBM Bluemix Service Description.

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Printed in USA