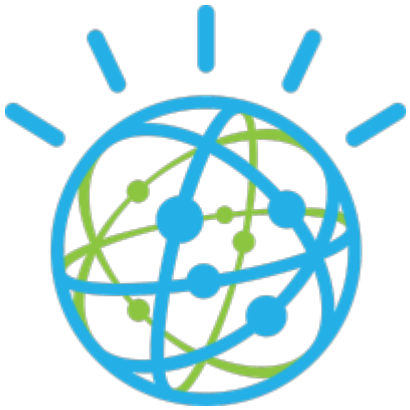
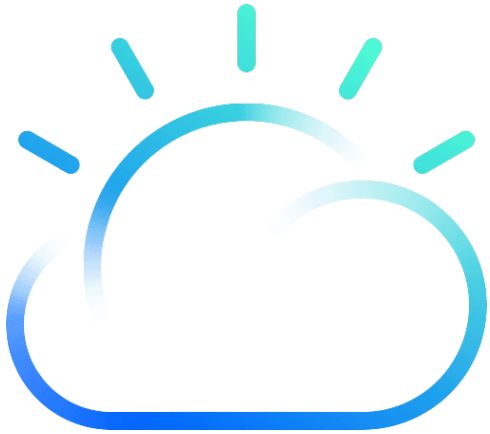
Hands-on Lab





IBM Cloud

Continuously deliver a chatbot app

Table of Contents

Introduction 4

Objectives 4

Pre-Requisites 4

Steps 5

Step 1 – Create a new web application 5

Step 2 – Create and bind the Conversation service 6

Step 3 – Configure the Conversation service 7

Step 4 – Enable Continuous Delivery for the app 8

Step 5 – Update the code 9

Step 6 – View the app 10

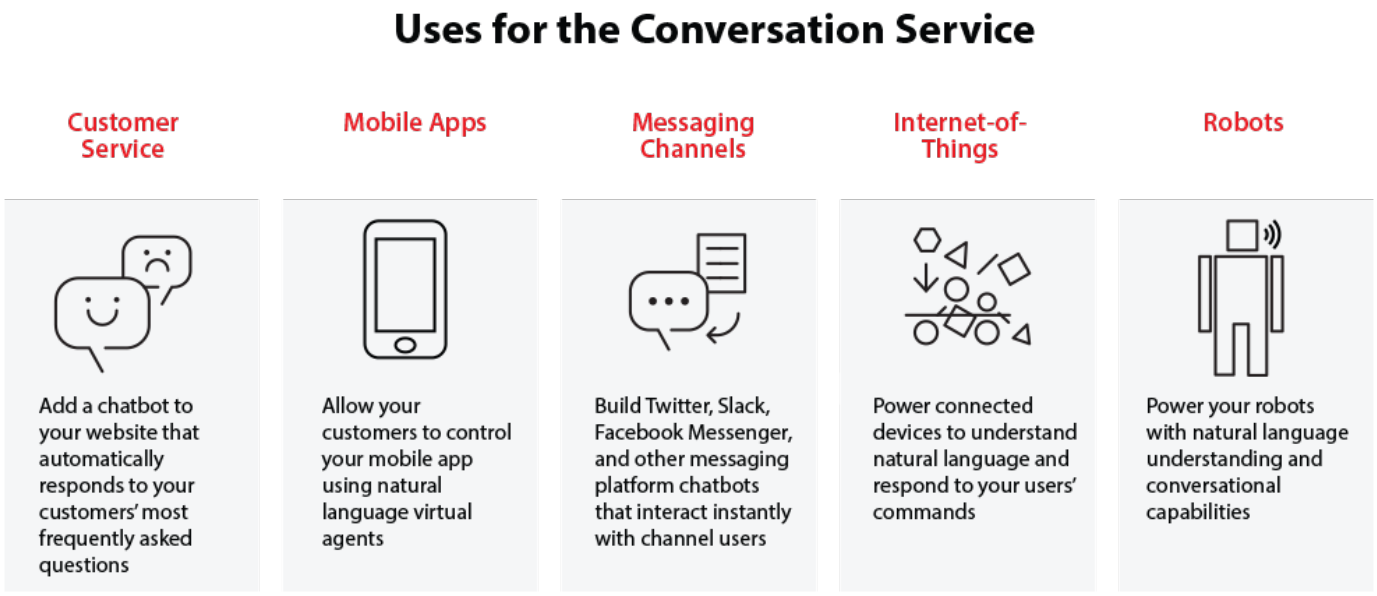
Appendix 10

IBM Cloud navigation tips 10

Resources 11

Introduction

With the IBM Cloud Conversation service you can create chatbots and virtual agents to combine machine learning, natural language understanding, and integrated dialog tools to provide automated customer engagements. The following image illustrates the variety of use cases possible with the Conversation service.



In this lab you'll gain a high-level understanding of the architecture, features, and development concepts of IBM Cloud by building a chatbot app utilizing a Node.js runtime bound to a Conversation service.

Objectives

In the following lab, you will learn:

- The basics of the IBM Cloud public service catalog
- How to deploy a new Cloud Foundry app based on a Node.js runtime
- How to create a Conversation service that works with an app
- The basics of the Conversation service and how it works
- How to integrate Continuous Delivery with an app

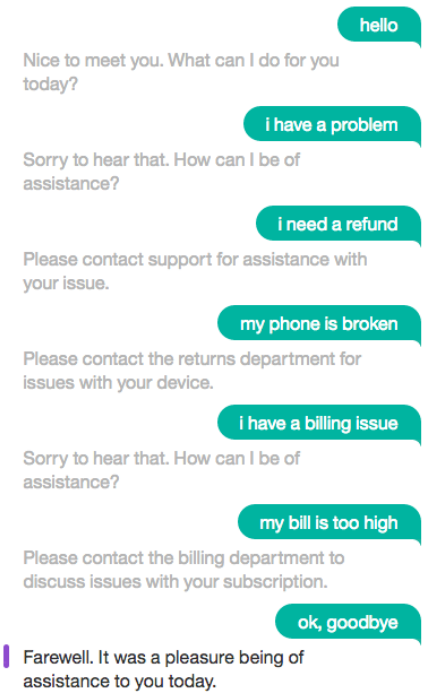
Pre-Requisites

- Get an IBM Cloud IBM id (https://bluemix.net/) or use an existing account

Steps

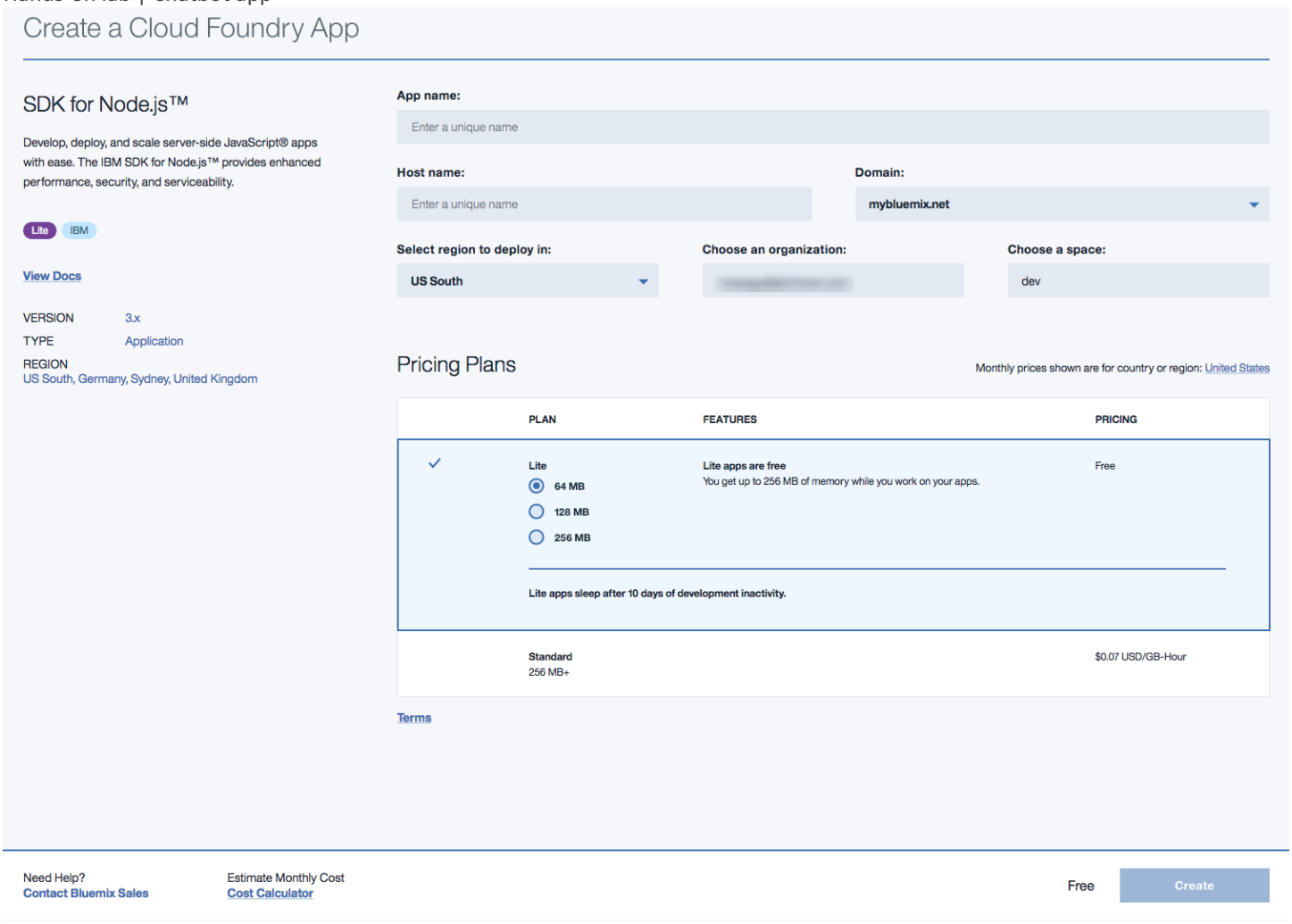
1. Create a new web application
2. Create and bind the Conversation service
3. Configure the Conversation service
4. Enable Continuous Delivery for the app
5. Update the code
6. View the app

Following these steps will result in a fully-functional chatbot app.



Step 1 – Create a new web application

First we need to create a runtime in which the app can run. Go through the following steps to create the runtime for the app and also assign it a hostname to make it accessible via the Internet.

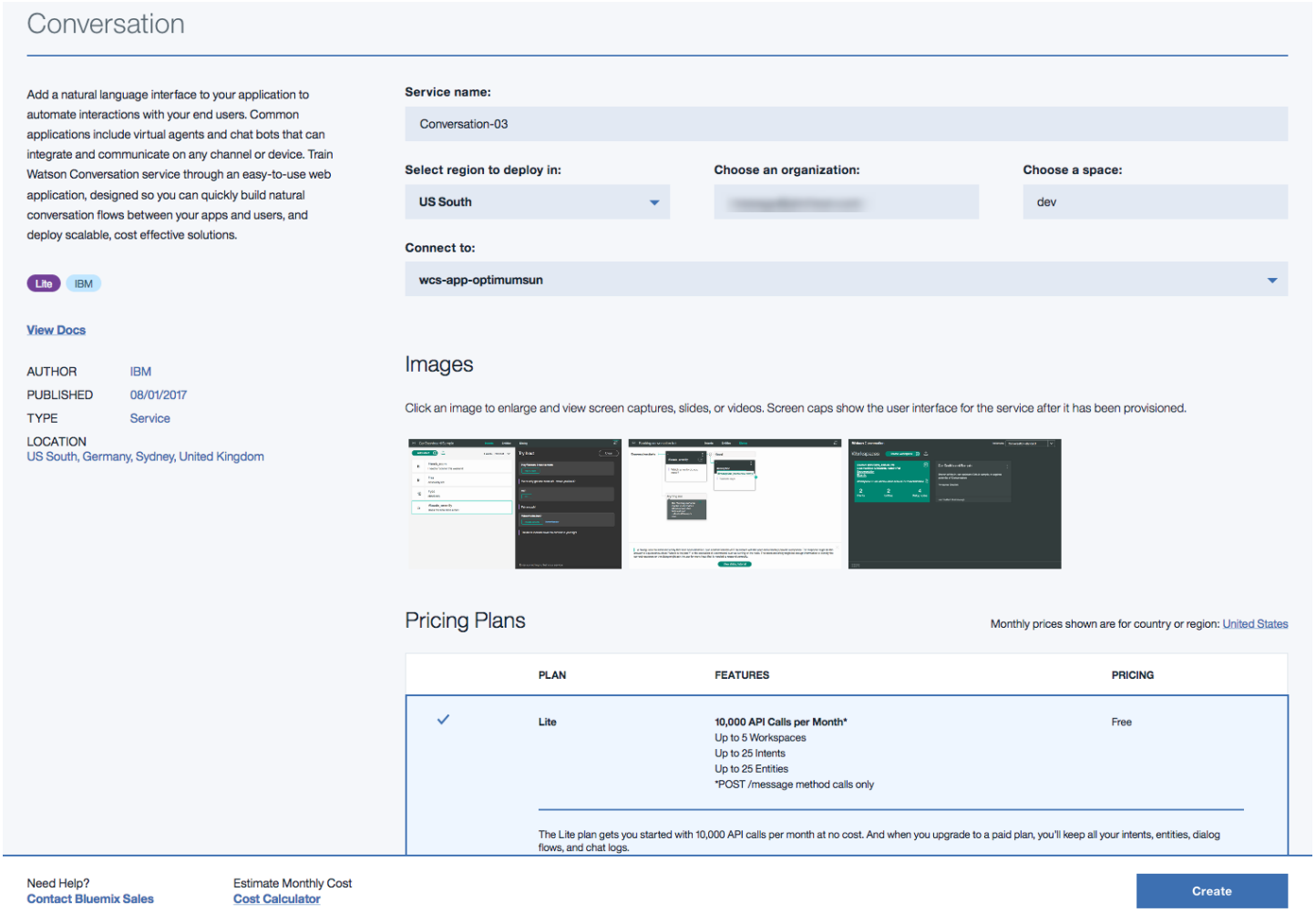


1. Log in to IBM Cloud (https://console.bluemix.net)
2. Select the Region **US South** to create your app – this is where your app is physically hosted
3. Navigate to the **Catalog** via the link in the upper right-hand menu
4. From the left-hand menu of categories select **Cloud Foundry Apps**
5. Select the **SDK for Node.js™** tile
6. In the details view that appears provide a unique name and host (e.g. wcs-app-[your-initials])
7. Click **Create** to create the application

The SDK for Node.js™ created a simple "Hello World!" web app that will become our starting point.

Step 2 – Create and bind the Conversation service

In order to create a chatbot we'll need to make use of the Conversation service available from the IBM Cloud catalog. Go through the following steps to create the Conversation service and bind it with your app.

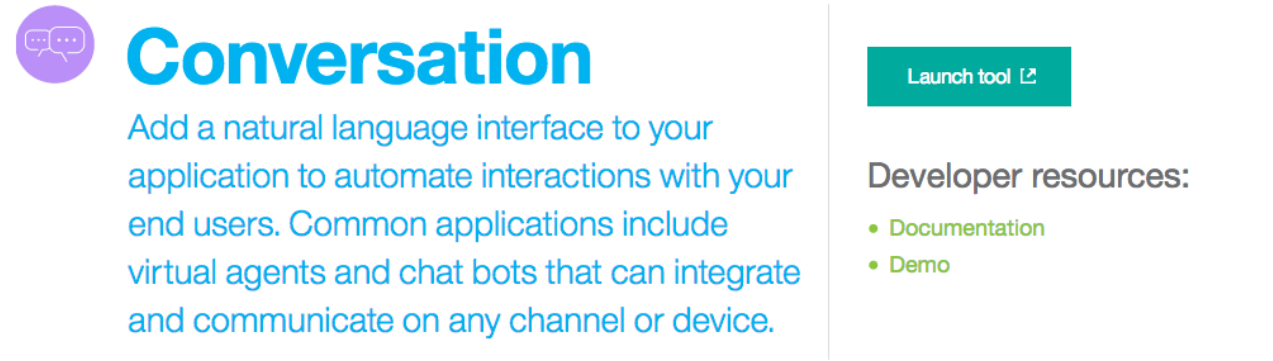


1. Navigate to the **Catalog** via the link in the upper right-hand menu
2. From the left-hand menu of categories select **Watson**
3. Select the **Conversation** tile
4. In the details view that appears select the same **space** as your Cloud Foundry app
5. Click **Create** to provision an instance of the Conversation service
6. Navigate to the Dashboard by clicking the hamburger icon in the top-left and selecting **Dashboard**
7. Click your app to view its **Overview**
8. Within the Connections box click **Create connection** to bind a service with your app
9. Highlight the row for the **Conversation** service and click **Connect**
10. Select **Restage** when prompted

Your application will restart and the service connection information will be made available for use by your application.

Step 3 – Configure the Conversation service

With the Conversation service provisioned and connected with your app, let's now configure it to carry out an actual conversation. To save time for the purposes of this lab you'll be importing an existing workspace configuration for use with your Conversation service. A workspace contains all the configuration information that gives a chatbot its unique personality and is a logical grouping of all the artifacts that define its behavior. Go through the following steps to configure your Conversation service.



1. Download the workspace JSON file and save it locally to your machine (https://raw.githubusercontent.com/IBM/watson-conversation-slots-intro/master/data/watson-pizzeria.json)
2. Back in the IBM Cloud **Dashboard** navigate to the **Manage** section of your Conversation service
3. Click on the **Launch tool** button to get started – the Workspaces view appears
4. Click the **Import workspace** button next to the Create button – the **Import a workspace** modal appears
5. Click the **Choose a file** control and select the previously downloaded JSON file
6. Click **Import** (keep the default selection of Everything) to import the workspace – the build view of the workspace appears
7. Click **Deploy** in the left-hand menu
8. Click the **Credentials** tab and copy the **Workspace ID** value – you'll need it in Step 5

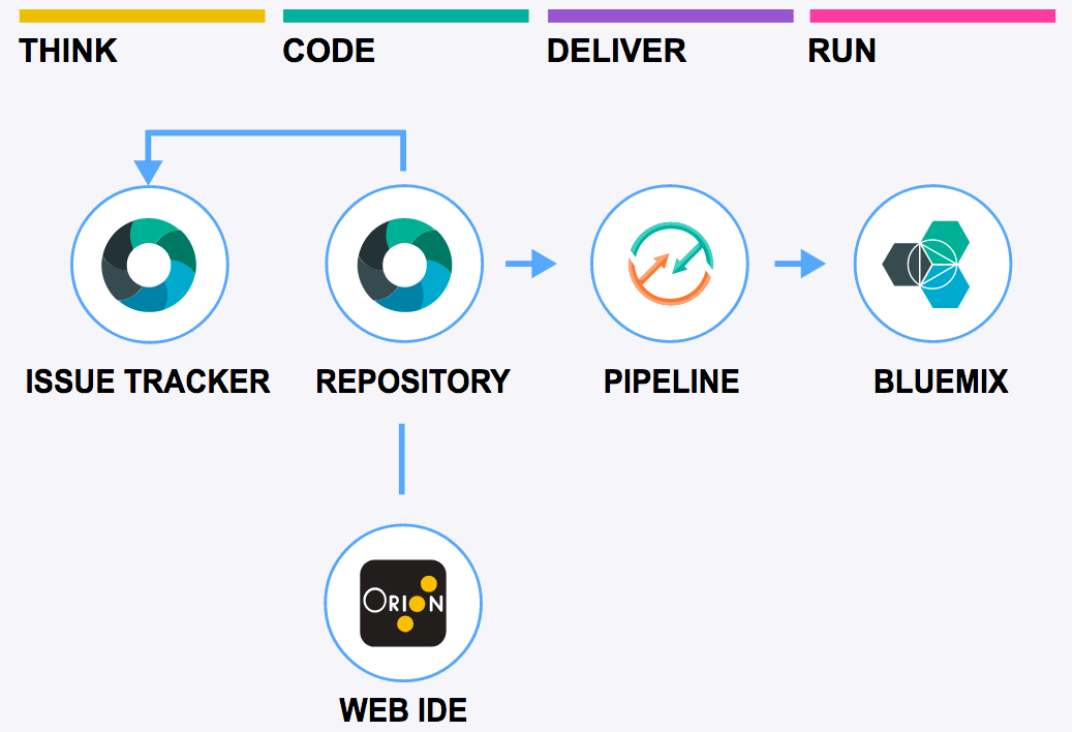
Information about Intents, Entities, and Dialog from the Build view of the workspace:

Intents represent the purpose of a user's inquiry, such as questions related to billing or defective products. Providing several examples of user input for a particular intent helps train the service to recognize each intent. Entities represent a term or object relevant to an intent and provides context-awareness for a specific intent. Providing synonyms and other possible values for an entity helps train the service to recognize a specific entity. A dialog defines the conversation flow based on the recognized intents and entities and encapsulates the appropriate responses for each.

For more information about configuring a Conversation workspace check out this link: https://console.bluemix.net/docs/services/conversation/configure-workspace.html#configuring-a-conversation-workspace.

Step 4 – Enable Continuous Delivery for the app

Now let's get the source code for the app that actually makes use of the Conversation service configured in the previous step. We can accomplish this quickly by utilizing the Continuous Delivery service of IBM Cloud to integrate an existing source code repository and setting up a toolchain to build and deploy the app.



1. Go to the **Overview** section of your app and click the **Enable** button under **Continuous delivery**
2. A new window opens to configure the toolchain – you can change the default name if desired
3. Under **Tool Integrations** at the bottom select **Git Repos and Issue Tracking**
4. Keep the **Repository type** as **Clone** and set the **Source repository URL** to https://github.com/optimumsun/conversation-simple.git
5. Click **Create** to create the toolchain which includes creation of a delivery pipeline that includes stages to build and deploy your app automatically upon every code commit

Step 5 – Update the code

With the code in place for our app let's make some updates to see the Continuous Delivery service in action. Complete the following steps to personalize the code and update a configuration file with a parameter specific to your instance of the Conversation service previously created.

1. Go to the **Overview** section of the toolchain and click the **Eclipse Orion Web IDE** tile – this will open up the web editor view for the code
2. Expand the file tree in the left-hand pane and click the **manifest.yml** file to edit it
3. Update the **name** value to match the name you gave for your Node.js app in Step 1
4. Replace the **my-conversation-service** keys on lines 3 and 13 with the name you gave your Conversation service
5. Underneath NPM_CONFIG_PRODUCTION add the following line, inserting the workspace ID copied earlier for the placeholder: WORKSPACE_ID: <YOUR-WATSON-CONVERSATION-WORKSPACE-ID>
6. From the file tree open the **public/index.html** file and edit line 4 to a title of your choosing
7. On the far left-hand vertical menu click the Git icon to bring up the Git display
8. Enter a commit message in the right-hand panel (e.g. updated manifest and index files)
9. Click **Commit** in the upper right-hand corner of the right pane to commit the altered files
10. Click **Push** in the left-hand panel to push the changes – this automatically triggers the app to rebuild and redeploy

11. Click the back arrow in the left-hand vertical menu to go back to the toolchain view
12. Click the **Delivery Pipeline** tile to view the Build and Deploy Stages
13. You'll see the Build and Deploy Stages automatically running – wait until the Deploy Stage completes

Holy cow! You now have a fully-functioning chatbot app hosted on IBM Cloud accessible via the Internet. Proceed to the next step to try out your app!

Step 6 – View the app

Now that we've built and deployed our app and integrated it with the Conversation service let's try it out!

1. Go to the **Overview** section of your Node.js app
2. When the status is running click the **Visit App URL** link to view the app
3. Start typing in the input box of your app to engage with your chatbot!

Congratulations! You've reached the end of this lab. To learn more about IBM Cloud and where to go from here consult the information found at the various links under the Resources section of this guide.

Appendix

IBM Cloud navigation tips

The following are a few points for navigating the IBM Cloud interface.

Viewing the Catalog

- Click Catalog in the upper right-hand menu to view the list of items within the IBM Cloud catalog

Viewing your list of items (Dashboard)

- Click the IBM Cloud text in the upper-left of the interface

Viewing the Overview of an app

- Navigate to your list of items
- Click on the item associated with your app – the Overview section should appear by default

Viewing Connections to your app

- Navigate to your list of items
- Click on the item associated with your app
- Click Connections in the left-hand menu to view the services associated with your app

Viewing your app URL

- Navigate to your list of items
- Click on the item associated with your app
- Click the Routes drop-down button in the upper-right – the hostname should appear

Viewing Service credentials

- Navigate to your list of items

- Click on the desired service
- Click on Service credentials in the left-hand menu
- Click on View credentials in the Actions column for the desired credentials to view

Viewing your list of Toolchains

- Click on the hamburger icon in the upper-left of the interface
- Click on the DevOps menu item – the list of Toolchains should appear by default

Resources

For additional information regarding IBM Cloud and the Watson Conversation service consult the following:

- IBM Cloud overview
  - https://console.bluemix.net/docs/overview/ibm-cloud.html#overview
- IBM Cloud case studies
  - https://www.ibm.com/cloud-computing/bluemix/case-studies
- IBM Cloud sample apps
  - https://ibm-bluemix.github.io/#!/
- developerWorks Recipes
  - https://developer.ibm.com/recipes/
- IBM Cloud Garage Method
  - https://www.ibm.com/devops/method
- Node.js library to access IBM Watson services
  - https://github.com/watson-developer-cloud/node-sdk
- Runtimes guides
  - https://www.ibm.com/blogs/bluemix/2017/03/runtimes-get-started-guides/
- Watson Accelerators
  - https://watsonaccelerators.mybluemix.net/welcome
- Watson Conversation Documentation
  - https://console.bluemix.net/docs/services/conversation/index.html#about