

IBM Training

Student Exercises

IBM Cognos Analytics

Hands-On

IBM Technical Enablement

© Copyright IBM Corp. 2018

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	BigInsights™	Cognos®
DataStage®	DB™	DB2®
developerWorks®	Domino®	FileNet®
Global Business Services®	GPFS™	Guardium®
IBM Business Partner®	IBM Watson™	IMS™
Informix®	InfoSphere®	Insight™
iSeries®	LanguageWare®	Lotus Notes®
Lotus®	Notes®	OmniFind®
Optim™	Power®	PowerHA®
Rational Team Concert™	Rational®	Redbooks®
SPSS®	System Storage®	System x®
System z®	Tivoli®	Watson™
WebSphere®	z/OS®	

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

VMware and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

Netezza® is a trademark or registered trademark of IBM International Group B.V., an IBM Company.

SoftLayer® is a trademark or registered trademark of SoftLayer, Inc., an IBM Company. Other product and service names might be trademarks of IBM or other companies.

January 2017 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright IBM Corp. 2018

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

Exercises description

This lab guide is intended to provide you with hands-on experience in using Watson Assistant to build a retail chatbot.

This course includes the following exercises:

- Exercise: Explore Cognos Reports and Visualizations
- Exercise: Upload data into Cognos
- Exercise: Explore Cognos data integration
- Exercise: Build a Dashboard
- Exercise: Embed the Cognos Dashboard into our Application

The exercises should be completed in the same order as listed before moving on to the next exercise.

Pre-requisites:

- a) An IBM Cloud account. The account is free and provides access to everything you need to develop, track, plan, and deploy apps. Sign up for a trial. The account requires an IBMid. If you don't have an IBM ID, you can create one when you register. Open up <https://www.ibm.com/cloud/>
- b) A Cognos account.

Note: The illustrating screenshots provided in this lab guide could be slightly different from what you see in the IBM Cloud interface you are using because of updates to IBM Cloud since the lab guide was created.

Exercise 01: Explore Cognos Reports and Visualizations.

Exercise Instructions

Lesson 1.1 Sign up for IBM Cognos and explore parts of the dashboard.

Pre-requisites:

- For this step, you should be able to log into IBM Cloud and navigate to your dashboard.
1. Open <https://www.ibm.com/account/us-en/signup/register.html?a=CognosAnalytics> in Chrome.
 2. Enter relevant information and click Continue.

Complete your profile by email.

The following information is required to access this application

Company * ✓

Are you a student? Yes No ✓

Country or region * (?) ✓

State or province * ✓

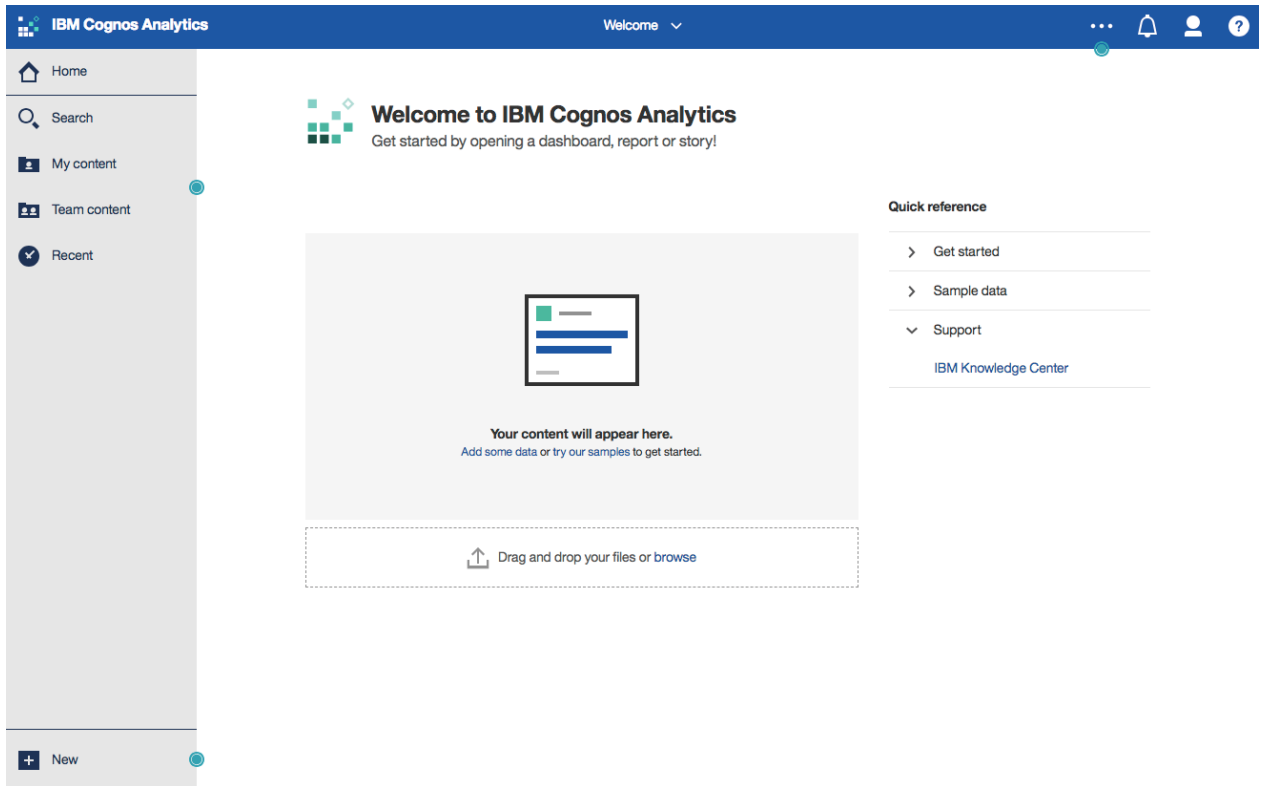
IBM may use my contact data to keep me informed of products, services and offerings:

You can withdraw your marketing consent at any time by sending an email to netsupp@us.ibm.com. Also you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each such email.

More information on our processing can be found in the [IBM Privacy Statement](#). By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement.

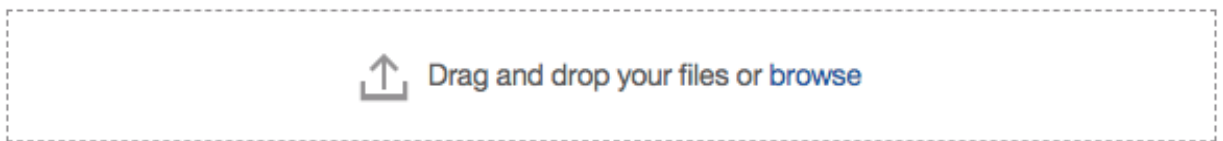
I accept the product [Terms and Conditions](#) of this registration form.

3. You should be greeted by the Cognos Analytics Dashboard.



4. Note the different parts of the home page contents.

Recent. These tiles will not be visible until content is added. The default view of the homepage is show tiles that represent the most recently modified content. The most recent six (6) are presented. The recent list may be expanded out to the most recent twenty (20) by selecting the Show more... Select Show less... to collapse the view back to the most recent six (6).



Drop Zone. Users may now easily upload files to Cognos Analytics using the Drag and drop your files or browse section at the bottom of the home page.



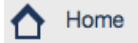
Quick Reference. Users will find resources to help them get started in Cognos Analytics

© Copyright IBM Corp. 2018

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

including videos, same data sets and links to the Cognos Analytics User Community and the IBM Knowledge Center.

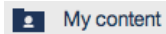
5. Note the different parts of the Navigation panel on the left.



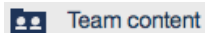
Home. The Home button appears in the top-left corner of the navigation panel. Users may return to this homepage at any time with the click of this button.



Search. Search. The New Smart Search in Cognos Analytics provides a modernized search engine that uses a smart, intent-driven search algorithm to assist the User. Click on Search to open the search panel. Type “Sales” in the search dialog box. As you type, an auto-fill feature will launch and render search suggestions for related terms. We will work more with this feature in an upcoming exercise. Click outside the Search panel to close it.



My Content. The My Content folder provides the User with direct access to the content they have saved. This is content owned by the User and may only be viewed by the User. You will be saving your work from today’s workshop in this folder. Click on My Content to open the navigation panel to see if there is any User content in your environment. Click outside the My Content panel to close it.



Team Content. The Team Content folder contains all the published enterprise and shared content the user has permissions to view. Click on Team Content to open the navigation panel. Notice there is a list of folders. We will go deeper into these later in the exercises. Click outside the Team Content panel to close it.



Recent. The Recent button shows the User the most recently used list of content, up to 20 objects (reports, dashboards, data modules, etc.). Objects appear in order based on most recently used. Once an object is viewed, it will move to the top of the list. Click on Recent to see what, if any, are the most recently used objects in your environment. Hover your mouse over the icon to the left of each object to identify the type of object. Click outside the Recent panel to close it.




New. The New button is used by Users to create new content. It is intent-driven, meaning that it allows Users to select what type of content they wish to create, and the Cognos Analytics UI will open the associated capabilities in the canvas. From here, Users may

create new Reports, Dashboards, Stories, Data Modules, access Other Companion Applications (legacy studios from previous versions of Cognos) and upload files.


 Report

 Dashboard

 Story

 Data module

 Other

 Upload files

6. Note the different parts of the application toolbar at the top.



Switcher Menu. The switcher menu in the center of the application toolbar provides a dropdown button that allows Users to easily move between the different objects they have worked with during their current session, without opening additional browser windows. (None will currently show as we have not opened any objects so far, but sample below shows example). The Switcher menu will display the name of the object currently active in the canvas.



More (3 horizontal ellipses). The More button provides the User with options to customize their User Experience. The options presented dynamically update based the type of object open in the canvas. Click outside the More menu to close it.



Notifications. Users may subscribe to Cognos Analytics content and receive notifications when that content has been updated and is ready for review. An indicator is provided to provide Users with the number of new notifications. Clicking on the Notifications button will open a listing of all unread notifications.

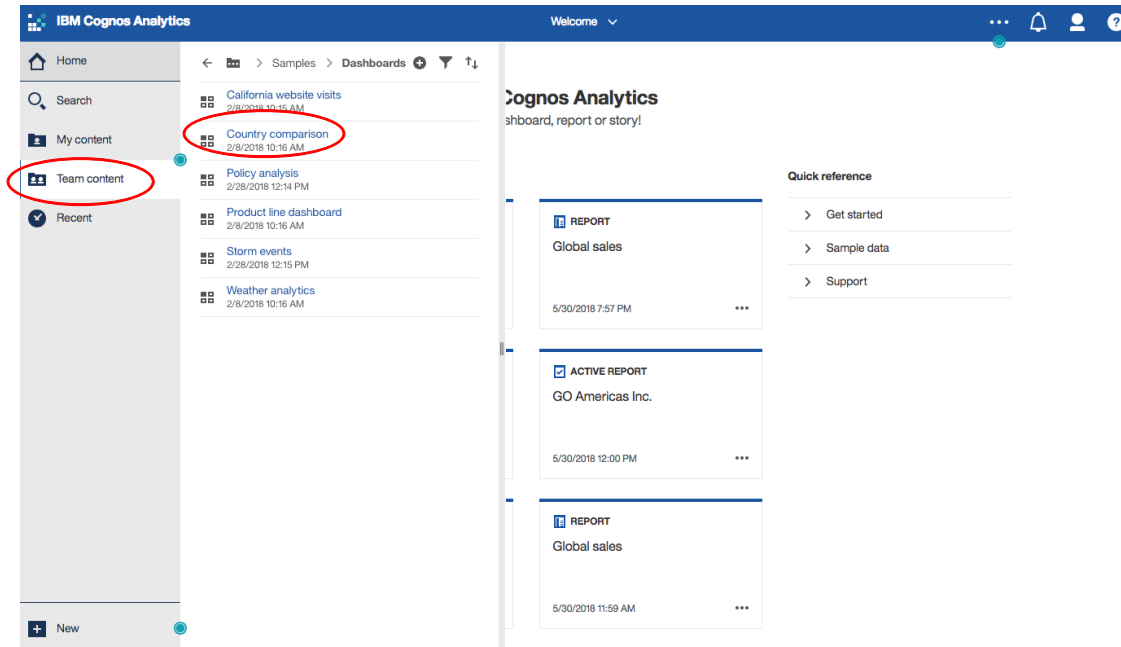


Personal Menu. The Personal Menu allows Users to change personal preferences for their environment and to manage their subscriptions. Click on the Personal Menu to see the capabilities available. Click outside the Personal Menu to close it.

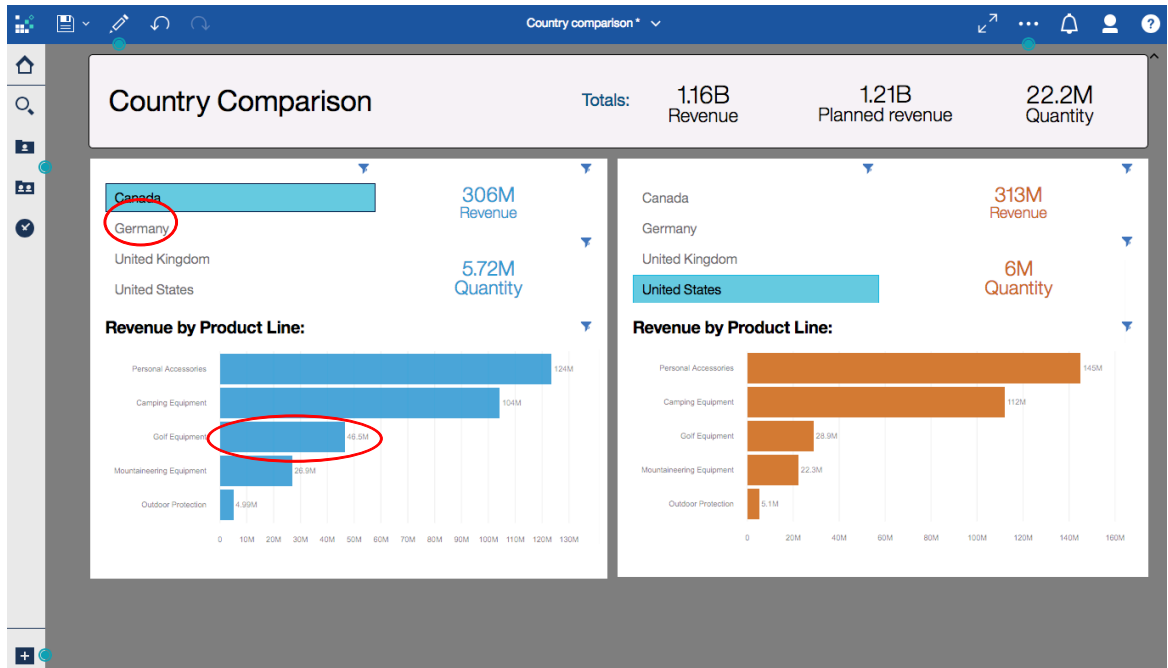


Help. The Help button allows users to access more information regarding Cognos Analytics.

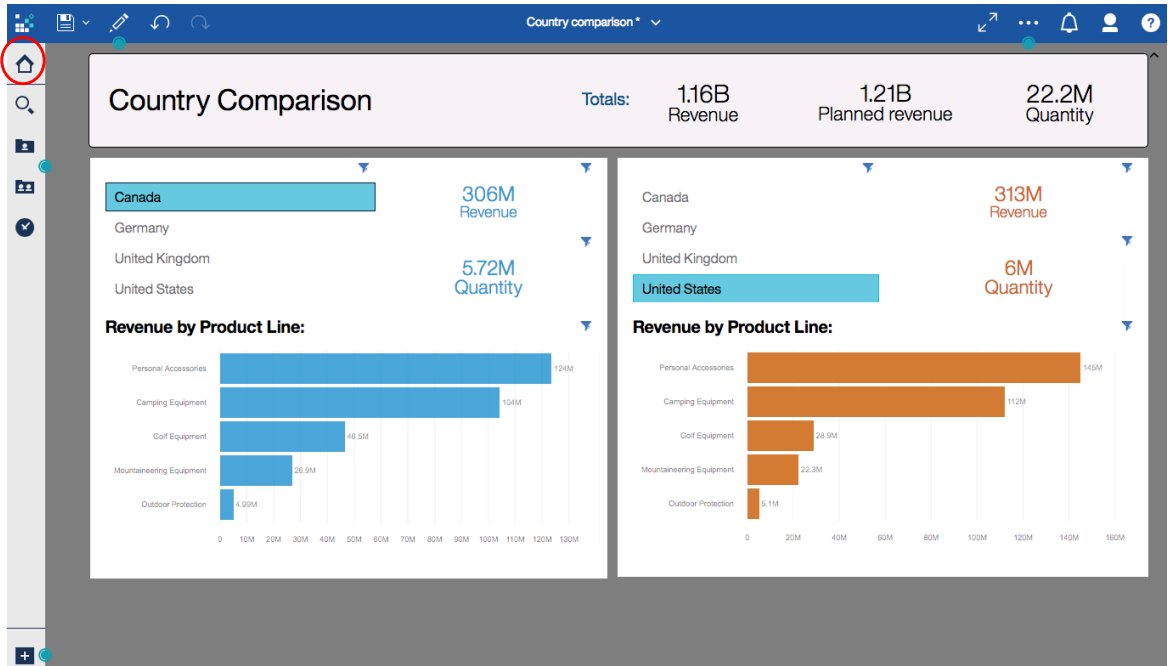
7. Click on team content. Navigate to Samples>Dashboards. Click on Country comparison.



8. This is a dashboard built for an outdoors equipment retailer to compare their sales between countries. Cognos dashboards are fully interactive. Click on list elements or on the graph to adjust the dashboard. Dashboards are embeddable in websites or applications or can be built in report form. Later in the lab, we will build our own dashboard and embed it into our web application.



9. Click on the Home button to return to your home page.



Lesson 1.3 Upload Data and Build a Data Module

1. Go to the IBM cloud dashboard <https://console.bluemix.net/dashboard/apps>.
2. Click on your watson-online-store app.

Cloud Foundry Apps 128 MB/2 GB Used

Name	Route	Memory (MB)	State
watson-online-store-20180324052725668	watson-online-store-...	128	Running (1/1)

3. You will be brought to the dashboard for your app.

The screenshot shows the IBM Cloud dashboard for a specific Cloud Foundry application. The top navigation bar includes the IBM Cloud logo and links for Catalog, Docs, Support, and Manage. The left sidebar lists navigation options: Getting started, Overview, Runtime, Connections, Logs, Monitoring, and API Management. The main content area displays the app's details: 'watson-online-store-20180324052725668' is in a 'Running' state. Below this, the 'Runtime' section features four circular gauges: 'BUILDPACK' (Python), 'INSTANCES' (1 instance, 100% health), 'MB MEMORY PER INSTANCE' (128 MB), and 'TOTAL MB ALLOCATION' (128 MB, with 1.875 GB still available). At the bottom, there are sections for 'Connections (3)' showing a connection to 'wos-cloudantNoSQLDB-service' and 'Runtime cost'.

4. From the menu on the left, click Runtime. Then Click the Environment variables tab.

The screenshot shows the IBM Cloud console interface. On the left, a navigation menu has 'Runtime' circled in red. The main content area shows the application 'watson-online-store-2018032...' in a 'Running' state. Below the application name, there are three tabs: 'Memory and instances', 'Environment variables' (circled in red), and 'SSH'. Under the 'Environment variables' tab, the 'VCAP_SERVICES' section is visible, containing a JSON configuration for a service named 'discovery' with its credentials.

5. Scroll down to the “User defined” section.

6. Set the value of SLACK_BOT_TOKEN to the API token you copied to your clipboard earlier.

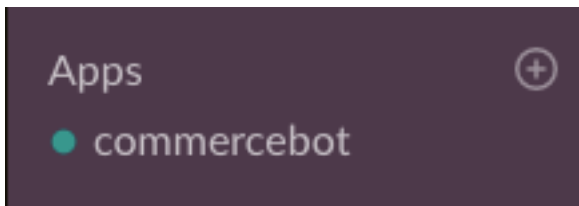
7. Set the value of SLACK_BOT_USER to the name of your bot.

The screenshot shows the 'User defined' section of the environment variables configuration. It features a table with three columns: NAME, VALUE, and ACTION. The table contains three rows of user-defined variables. Below the table are four buttons: Add, Save, Reset, and Export.

NAME	VALUE	ACTION
CLOUDANT_DB_NAME	watson-online-store	⊗
SLACK_BOT_TOKEN	xoxb-336532171303-fqgGDpvPjKbELXfHzGtuDwn!	⊗
SLACK_BOT_USER	commercebot	⊗

8. Click Save. Your application will automatically restart.

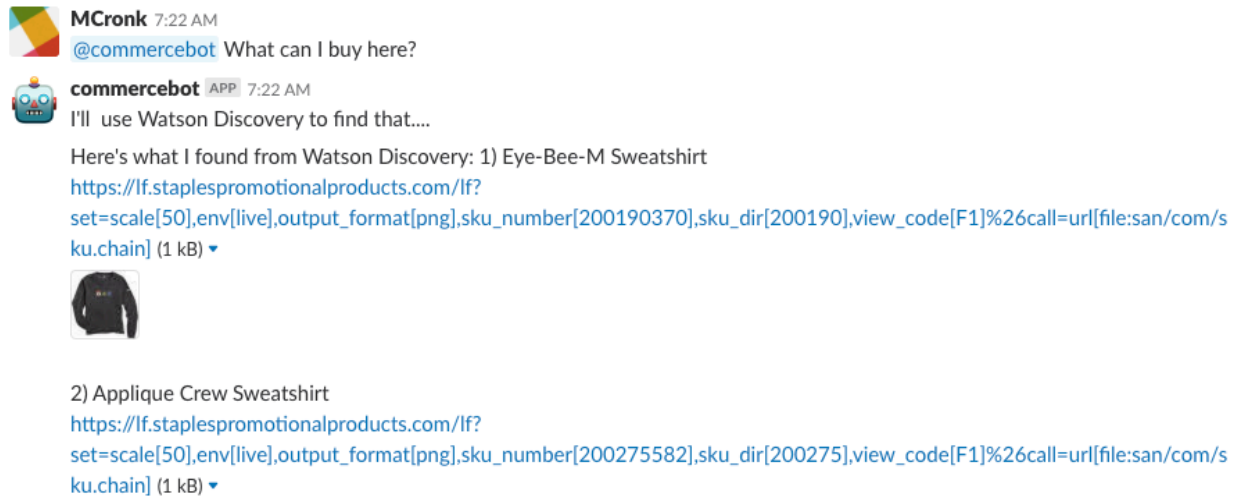
9. Once the application is running, you should also check your bot status on Slack. When the application is running and the bot status is green/online, then you can chat.



Lesson 1.4 Chat with your bot.

Pre-requisites:


- Completion of Lessons 1.1 – 1.3.
1. Go to the Slack channel that you created. Your bot user should be a member of the channel. Start a conversation in the channel by typing “@<name-of-your-bot> What can I buy here?” in the chat window. The bot returns a partial list of items.



MCronk 7:22 AM
@commercebot What can I buy here?

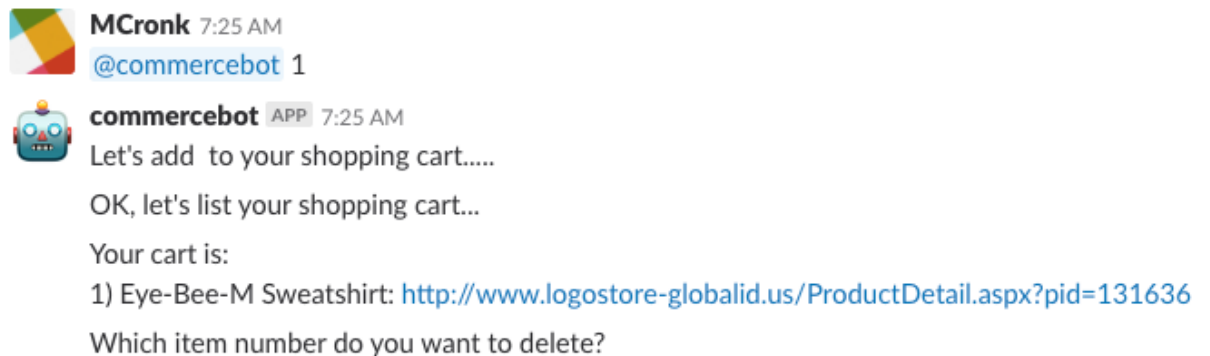
commercebot APP 7:22 AM
I'll use Watson Discovery to find that...

Here's what I found from Watson Discovery: 1) Eye-Bee-M Sweatshirt
[https://lf.staplespromotionalproducts.com/lf?set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200190370\],sku_dir\[200190\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200190370],sku_dir[200190],view_code[F1]%26call=url[file:san/com/sku.chain]) (1 kB) ▾



2) Applique Crew Sweatshirt
[https://lf.staplespromotionalproducts.com/lf?set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200275582\],sku_dir\[200275\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200275582],sku_dir[200275],view_code[F1]%26call=url[file:san/com/sku.chain]) (1 kB) ▾

2. Choose an item to add to your cart by typing the item number. For example, to add the Eye-Bee-M Sweatshirt to your cart, type @<name-of-your-bot> 1.




MCronk 7:25 AM
@commercebot 1


commercebot APP 7:25 AM
Let's add to your shopping cart....
OK, let's list your shopping cart...
Your cart is:
1) Eye-Bee-M Sweatshirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=131636>
Which item number do you want to delete?

3. When the bot asks “Which item number do you want to delete?”, you can respond with a 0 to continue. Add a few more items to your cart. The online store lists your cart every time you add or remove an item.

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

 **MCronk** 7:28 AM
@commercebot 0


 **commercebot** APP 7:28 AM
Let's Delete 0.


Now we'll list the cart...



7:28 AM Items in your cart are now:

1) Eye-Bee-M Sweatshirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=131636>

 **MCronk** 7:29 AM
@commercebot What mugs can i buy?

 **commercebot** APP 7:29 AM
I'll use Watson Discovery to find that....

Here's what I found from Watson Discovery: 1) IBM C-Handle Mug 11oz.

[https://lf.staplespromotionalproducts.com/lf?](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200245419],sku_dir[200245],view_code[F1]%26call=url[file:san/com/sku.chain](998 B) ▾)

[set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200245419\],sku_dir\[200245\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]\(998 B\) ▾](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200245419],sku_dir[200245],view_code[F1]%26call=url[file:san/com/sku.chain](998 B) ▾)



Lesson 1.5 Update the Python code to add the checkout function and redeploy the app.

Pre-requisites:

- Completion of Lessons 1.1 – 1.4.
1. Go to the IBM cloud dashboard <https://console.bluemix.net/dashboard/apps>.
 2. Click on your watson-online-store app.

Cloud Foundry Apps 128 MB/2 GB Used

Name ^	Route	Memory (MB)	State
watson-online-store-20180324052725668	watson-online-store-...	128	Running (1/1)

3. You will be brought to the dashboard for your app.

The screenshot shows the IBM Cloud dashboard for a specific Cloud Foundry application. The top navigation bar includes 'IBM Cloud', 'Catalog', 'Docs', 'Support', and 'Manage'. The left sidebar lists navigation options: 'Getting started', 'Overview', 'Runtime', 'Connections', 'Logs', 'Monitoring', and 'API Management'. The main content area displays the app name 'watson-online-store-2018032...' with a '.py' icon and a 'Running' status. Below this, it shows 'Org: michael.cronk139@gmail.com', 'Location: US South', and 'Space: dev'. The 'Runtime' section features four circular gauges: 'BUILDPACK Python', 'INSTANCES 1 (All instances are running, Health is 100%)', 'MB MEMORY PER INSTANCE 128', and 'TOTAL MB ALLOCATION 128 (1.875 GB still available)'. At the bottom, there are sections for 'Connections (3)' showing 'wos-cloudantNoSQLDB-service' and 'Runtime cost'.

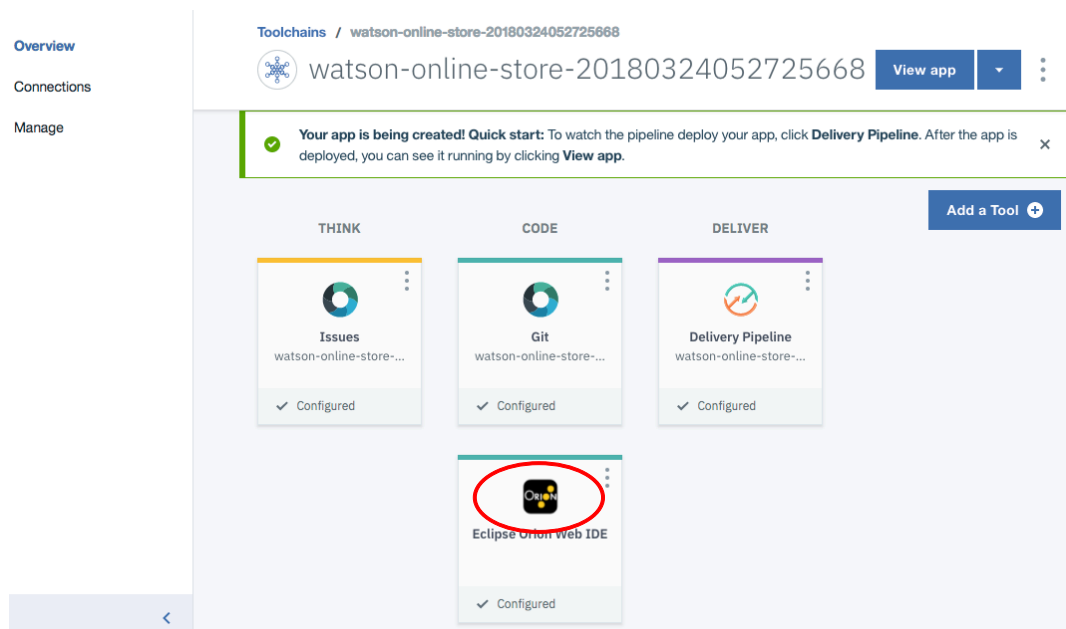
4. Scroll down and click “View Toolchain.”

Continuous delivery

You enabled continuous delivery and have a toolchain. With your toolchain, you can automate builds, tests, deployments, and more. [View Docs.](#)

[View toolchain](#)

5. Review the toolchain. In the toolchain, click Eclipse Orion Web IDE to open the editor and make changes to the Python code. From the left navigation menu of the editor, expand the watsononlinestore folder and click watson_online_store.py.



Note: Python is particular about indentation. Copy and paste might not work properly when you move the code to the cards. If you see errors that indicate indentation problems in the log on the IBM Cloud App Dashboard for your app, type the code by hand, being sure to comply with Python indentation rules.

6. Add the following code, including a blank line, after line 810. This code calls the checkout handler when checkout is requested in the chatbot.

```
if ('shopping_cart' in self.context.keys()) and
    self.context['shopping_cart'] == 'checkout':
    return self.handle_checkout()
```

The screenshot shows the IBM Cloud IDE interface. On the left, a file explorer shows the project structure for 'watson-online-store-20180517143342295'. The file 'watson_online_store.py' is selected and highlighted with a red circle. On the right, the code editor displays the Python code for 'watson_online_store.py'. The code includes a function 'handle_checkout()' and a 'def run(self):' block. Line 811 is highlighted with a blue background, showing the code to be added: 'if ('shopping_car' in self.context.keys() and self.context['shopping_cart'] == 'checkout'): return self.handle_checkout()'. The code editor also shows a menu bar with 'File', 'Edit', 'View', and 'Tools', and a status bar at the bottom indicating 'Line 811 : Column 1'.

7. Add the following code at line 821 above “def run(self):”. This code processes the checkout by removing all items from the cart in the Cloudant database.

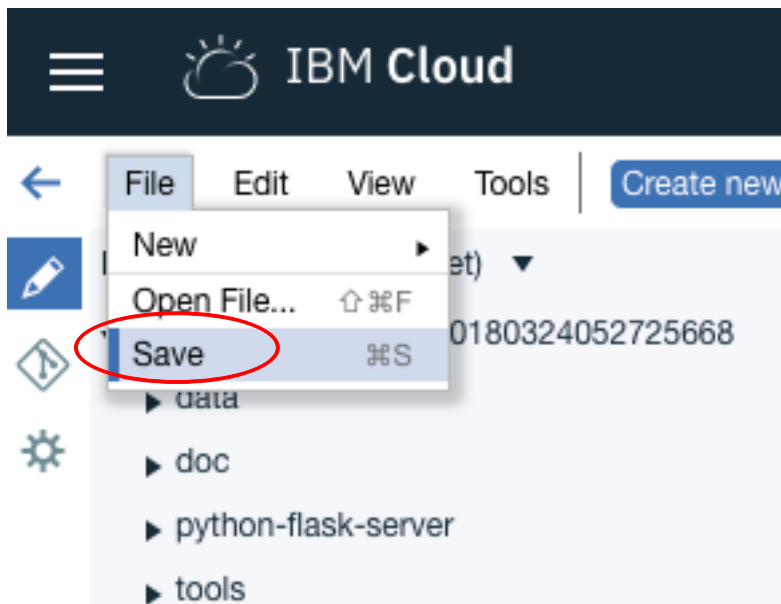
```
def handle_checkout(self):
    """ Removes all items from the cart """
    email = self.customer.email
    shopping_list = self.cloudant_online_store.list_shopping_cart(email)
    for _index, item in enumerate(shopping_list):
        self.cloudant_online_store.delete_item_shopping_cart(email, item)
    self.clear_shopping_cart() # this actually clears the context
    return False
```

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

```
watson_online_store.py
794
795     self.context['cart_item'] := {}
796     return self.handle_delete_from_cart()
797
798     if self.context.get('get_input') == 'no':
799         return False
800
801     return True
802
803 def handle_conversation(self, message, sender, user):
804     """Handler for messages coming from user.
805
806     Loops when additional input is needed.
807
808     :param str message: text from UI
809     :param sender: a sender impl used for send_message
810     :param str user: user ID
811     """
812     if ('shopping_car' in self.context.keys() and self.context['shopping_cart'] == 'checkout'):
813         return self.handle_checkout()
814
815     if user and not self.customer:
816         self.init_customer(sender, user)
817
818     get_input = self.handle_message(message, sender)
819     while not get_input:
820         get_input = self.handle_message(message, sender)
821
822 def handle_checkout(self):
823     """ Removes all items from the cart"""
824     email = self.customer.email
825     shopping_list = self.cloudant_online_store.list_shopping_cart(email)
826     for _index, item in enumerate(shopping_list):
827         self.cloudant_online_store.delete_item_shopping_cart(email, item)
828     self.clear_shopping_cart() # this actually clears the context
829     return False
830
831 def run(self):
832     """Main run loop of the application with a Slack client
833     """
834     # make sure DB exists
835     self.cloudant_online_store.init()
836
837     if self.slack_client and self.slack_client.rtm_connect():
838         LOG.info("Watson Online Store bot is connected and running!")
839         while True:
840             slack_output = self.slack_client.rtm_read()
841             if slack_output:
842                 LOG.debug("slack output\n:{}".format(slack_output))
843
844             message, channel, user = self.parse_slack_output(slack_output)
845             if message:
846                 LOG.debug("message:\n {} channel:\n {}" %
```

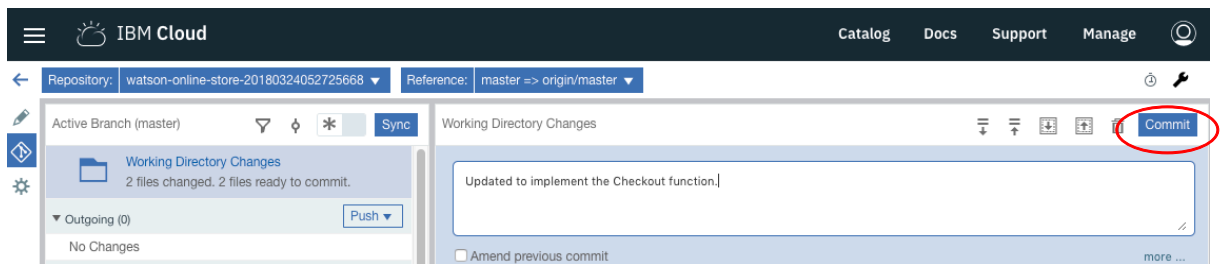
8. From the file menu click Save.



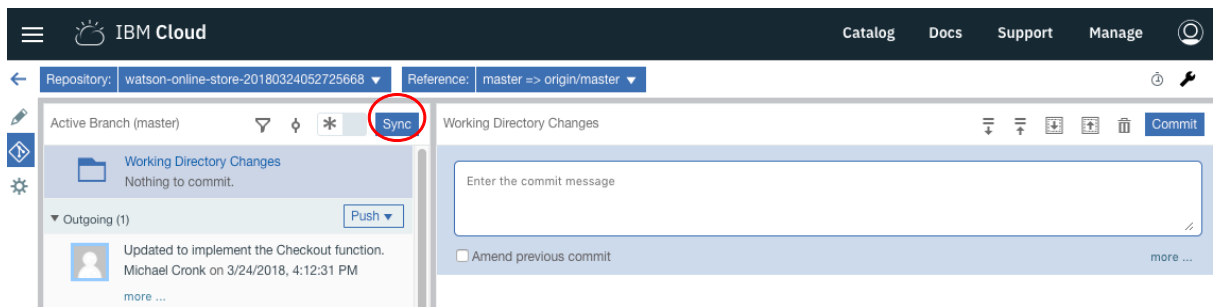
9. Open Git by clicking the Git icon below the pencil icon from the left navigation.



10. In the Working Directory Changes section, the changes to the Python file are shown. Type a commit message, such as “Updated to implement the Checkout function” and click Commit.

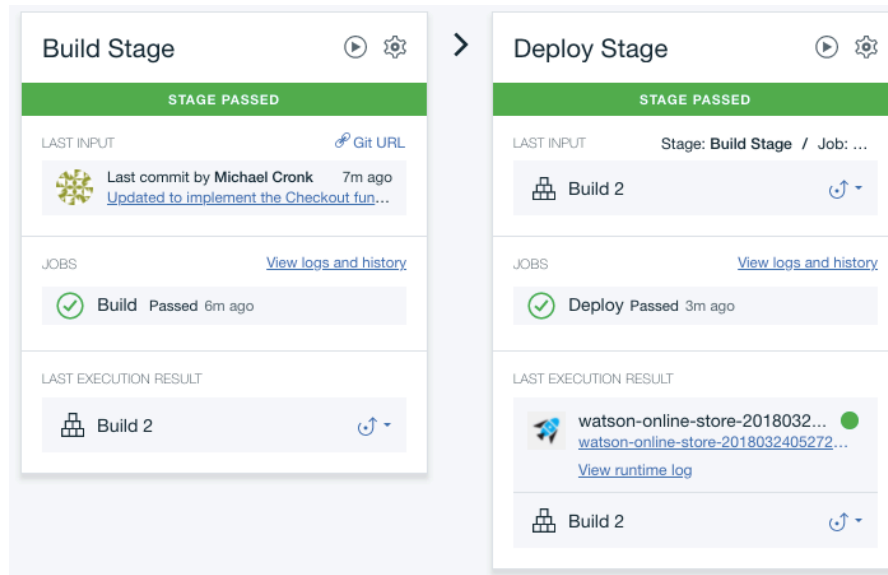


11. Click Sync to push the code. You can use the back browser arrow to go back to the toolchain.



12. Click on “Delivery Pipeline”

13. Because of the way the pipeline is configured, when the sync is complete, the pipeline automatically runs to build and deploy the changes to the app. Watch the pipeline until it completes the deploy stage. When it is finished, your modified app is up-and-running on IBM Cloud.



Lesson 1.6 Update the Python code to add the checkout function and redeploy the app.

Pre-requisites:

- Completion of Lessons 1.1 – 1.5.

14. Go to the IBM cloud dashboard <https://console.bluemix.net/dashboard/apps>.

15. Click on your watson-online-store app.

Cloud Foundry Apps 128 MB/2 GB Used

Name ^	Route	Memory (MB)	State
watson-online-store-20180324052725668	watson-online-store-...	128	● Running (1/1)

16. You will be brought to the dashboard for your app. On the left-side navigation panel, click on Connections.

IBM Cloud

Cloud Foundry apps /

watson-online-store-20180324052725668 ● Running

Org: michael.cronk139@gmail.com Location: US South Space: dev

Runtime

- BUILDPACK: Python
- INSTANCES: 1 (All instances are running, Health is 100%)
- MB MEMORY PER INSTANCE: 128
- TOTAL MB ALLOCATION: 128 (1.875 GB still available)

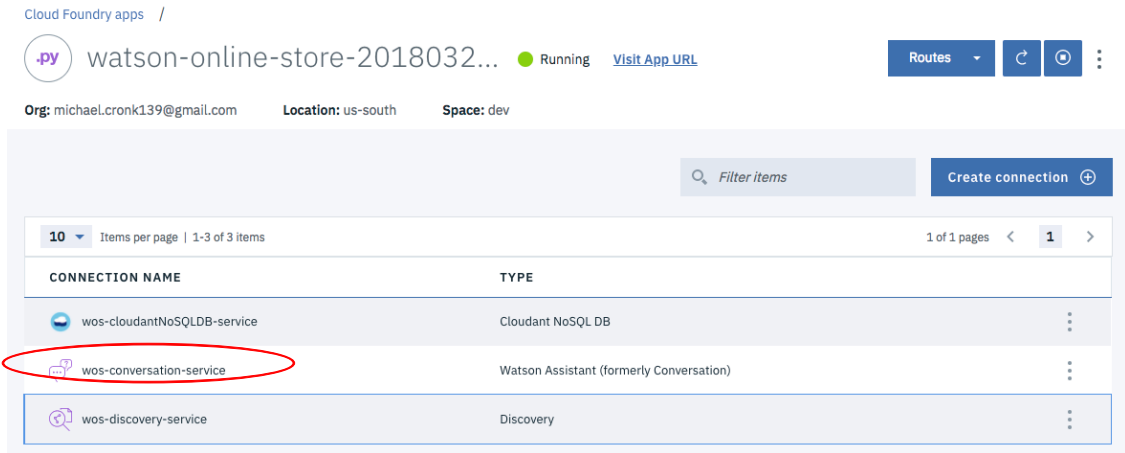
Connections (3)

- wos-cloudantNoSQLDB-service

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

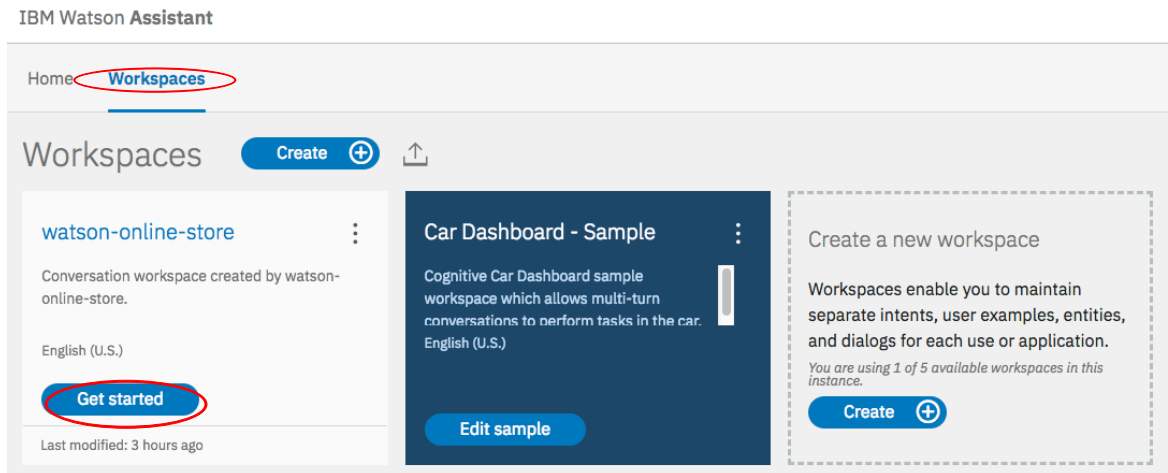
17. Click on your conversation service.



18. Click Launch Tool to open the Watson Assistant page.

19. Click on the Workspaces tab.

20. Click on your app workspace "watson-online-store."



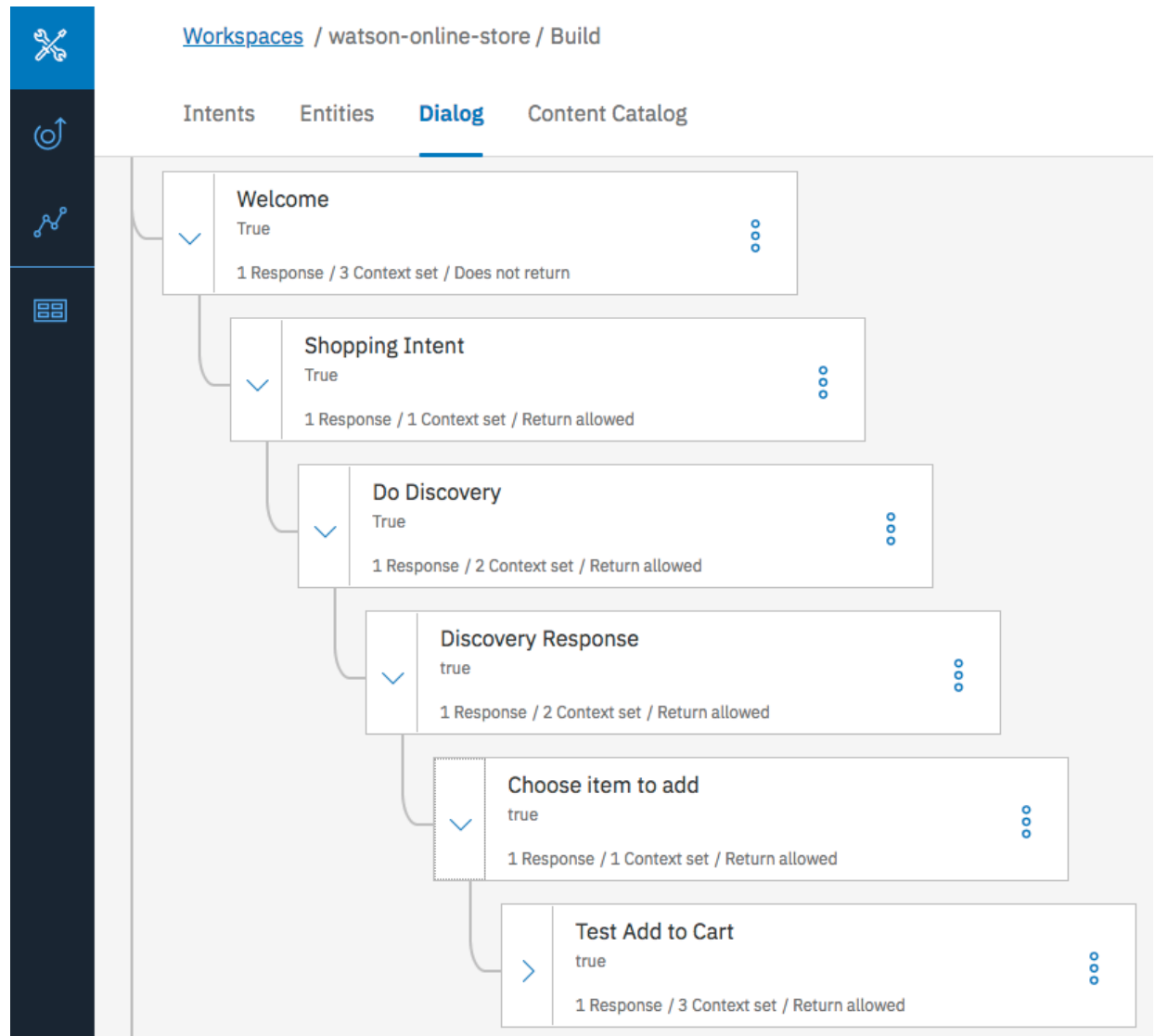
21. Click on the Dialog tab.

22. Click on the blue arrows on the left side of the nodes to expand them.

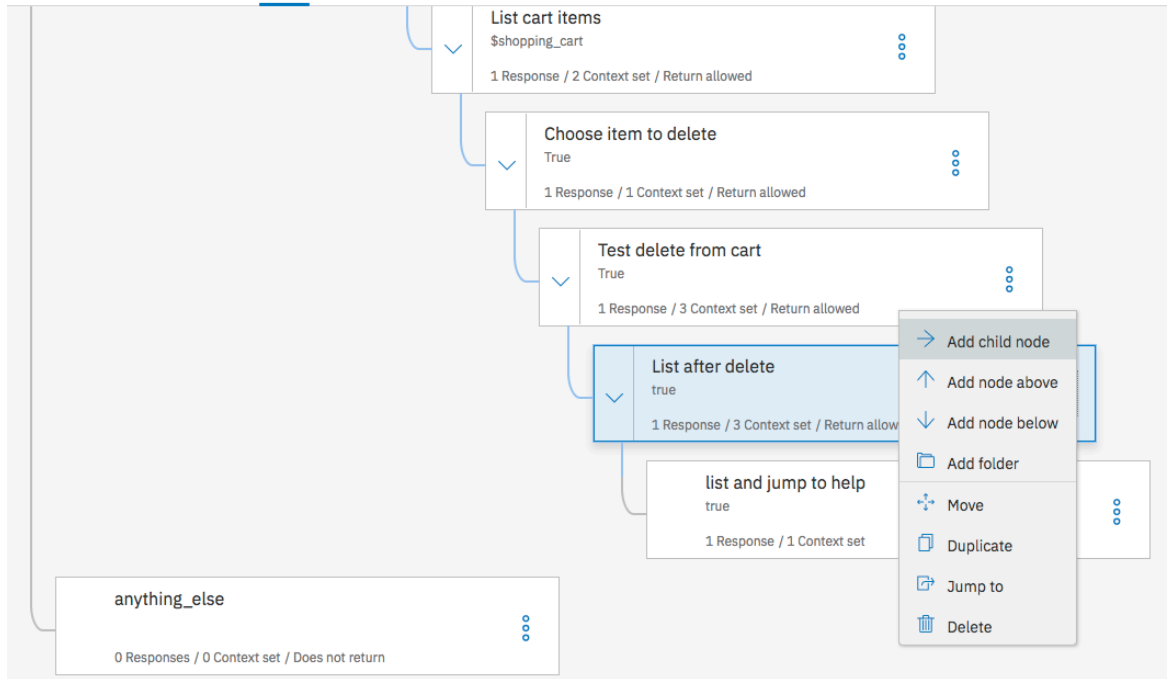
23. You will be making changes to the configured dialog to allow for the processing of Checkout by the bot. The changes will occur in and after the "list after delete" in

conversation configuration. Open the configuration by clicking the arrows on the nodes to expand the dialog until you see the "List after delete" node. You will be adding new nodes and several jumps in the conversation configuration. As you review the figure below, note that you will be adding everything that occurs after the "List after delete" node.

Note: Copy and paste might not work when you move the code to the nodes. If you see errors in the code entry box after you copy and paste the code, type the code by hand.



24. Click on the three dots on the right side of the "List after delete node" and then click Add child node.



25. For the name of the new node, type “list items”

26. Type “true” in the If bot recognizes field.

27. In the “Then respond with” section, click the menu icon next to Add response condition and select Open JSON editor.

list items

Customize



If bot recognizes:

true

Then respond with:

Enter a response...

Open JSON editor

Open context editor

28. Replace the existing JSON with the following code:

```
{
  "context": {
    "get_input": "no"
  },
  "output": {
    "text": {
      "values": [
        "Items in your cart are now:\n$shopping_cart"
      ],
      "selection_policy": "sequential"
    }
  }
}
```

list items

 Customize



If bot recognizes:

true  

Then respond with:



```
1 {
2   "context": {
3     "get_input": "no"
4   },
5   "output": {
6     "text": {
7       "values": [
8         "Items in your cart are now:\n$shopping_cart"
9       ],
10      "selection_policy": "sequential"
11     }
12   }
13 }
14
```

29. Add a “checkout node after the “list items” node by adding another child node.

30. For the name, type “Checkout”

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM

31. Type true in the If bot recognizes field.
32. In the “Then respond with” section, click the menu icon next to add response condition and select Open JSON editor.
33. Add the following code to the response:

```
{
  "output": {
    "text": {
      "values": [
        "Would you like to checkout and clear the cart?"],
      "selection_policy": "sequential"
    }
  },
  "context": {
    "cart_item": "",
    "get_input": "yes"
  }
}
```

Checkout

 Customize



If bot recognizes:

true  

Then respond with:



```
1 {
2   "output": {
3     "text": {
4       "values": [
5         "Would you like to checkout and clear the cart?"],
6       "selection_policy": "sequential"
7     }
8   },
9   "context": {
10    "cart_item": "",
11    "get_input": "yes"
12  }
13 }
14 |
```

After the checkout node, you will be adding two nodes which are used to respond to the positive response to checkout and the negative response indicating that the customer wants to continue shopping.

34. Add a “Clear the Cart” child node after “Checkout”

35. For the name, type “Clear the cart”

36. Type @PositiveResponse in the If bot recognizes field.

37. In the “Responses” section, click the menu icon and select JSON.

38. Add the following code to the response:

```
{
  "output": {
    "text": {
      "values": [
        "Ok, let's clear your cart."
      ],
      "selection_policy": "sequential"
    }
  },
  "context": {
    "get_input": "no",
    "shopping_cart": "checkout"
  }
}
```



If bot recognizes:

@PositiveResponse  

Then respond with:



```
1 {
2   "context": {
3     "get_input": "no",
4     "shopping_cart": "checkout"
5   },
6   "output": {
7     "text": {
8       "values": [
9         "Ok, let's clear your cart."
10      ],
11     "selection_policy": "sequential"
12   }
13 }
14 }
```

39. Add a “Keep shopping” child node after “Checkout”
40. For the name, type “Keep shopping”
41. Type @NegativeResponse in the If bot recognizes field.
42. In the “Responses” section, click the menu icon and select JSON.
43. Add the following code to the response:

```
{
  "output": {
    "text": {
      "values": [
        "Ok, let's keep shopping!"
      ],
      "selection_policy": "sequential"
    }
  },
  "context": {
    "get_input": "no"
  }
}
```

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM



If bot recognizes:

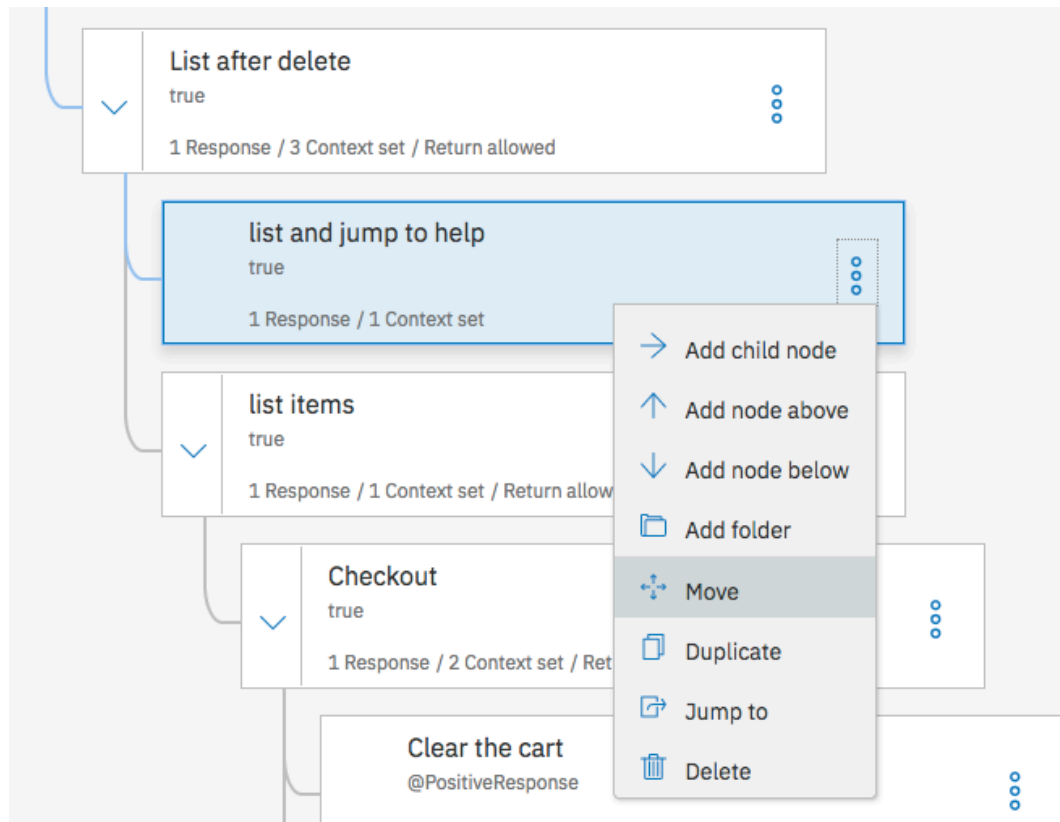
@NegativeResponse

Then respond with:



```
1 {
2   "context": {
3     "get_input": "no"
4   },
5   "output": {
6     "text": {
7       "values": [
8         "Ok, let's keep shopping!"
9       ],
10      "selection_policy": "sequential"
11    }
12  }
13 }
```

44. When a user wants to keep shopping, the cart is listed and they can continue shopping. You can do this by moving the "List and jump to help" node to after the negative response to the right of the "Keep shopping" card. To do this, you click the menu icon in the "List and jump to help" node and then click Move. A banner at the top of the screen says Select where you want to move the node to.



45. Click the “Keep shopping” node and then click As child node.
46. When the user has selected to clear the cart, they will want to go back to shopping. To allow this, you will add a "Go back to shopping" card followed by a jump that will take the user back to the place in the conversation where they can add more items to the cart.
47. Add a “Go back to shopping” child node after “Clear the cart”
48. For the name, type “Go back to shopping”
49. Type true in the If bot recognizes field.
50. In the list of Responses, type “Shopping cart has been cleared!”

Go back to shopping

 Customize



If bot recognizes:

true  

Then set context:



Variable

Value

 Add variable

Add value



 Add variable

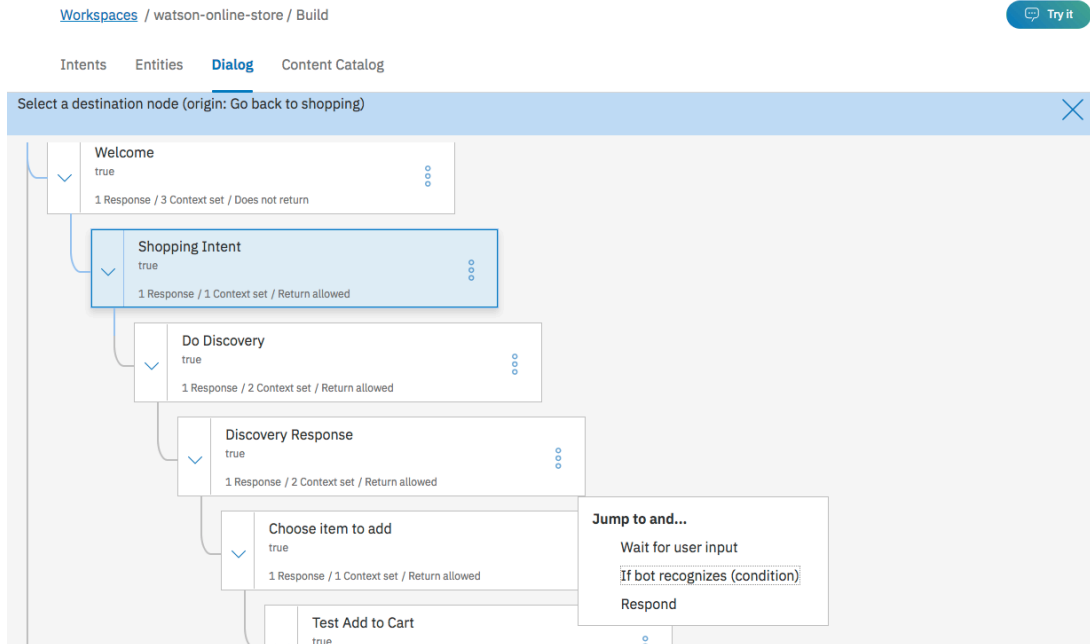
And respond with:

Shopping cart has been cleared!

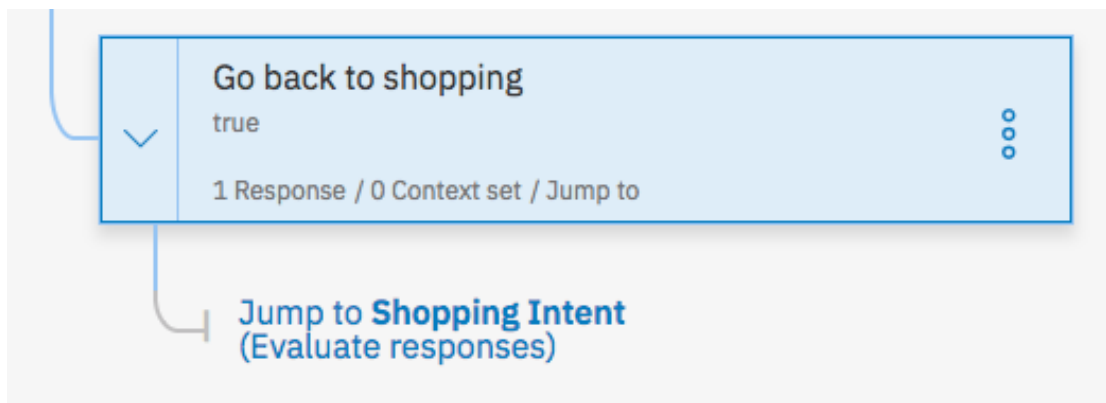
51. Add a “Jump to Shopping Intent” node after the “Go back to shopping” node. This node takes you back to the start of a new conversation with the bot.
52. Click the menu (three dots on the right side of the node) on the “Go back to shopping cart” node and then click Jump to. A blue banner at the top of the screen says “Select a destination node (origin: Go back to shopping)”
53. Scroll to the beginning of the conversation, click the “Shopping Intent” node, and then click Respond.

© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM



54. A card like the following example is added to the conversation.

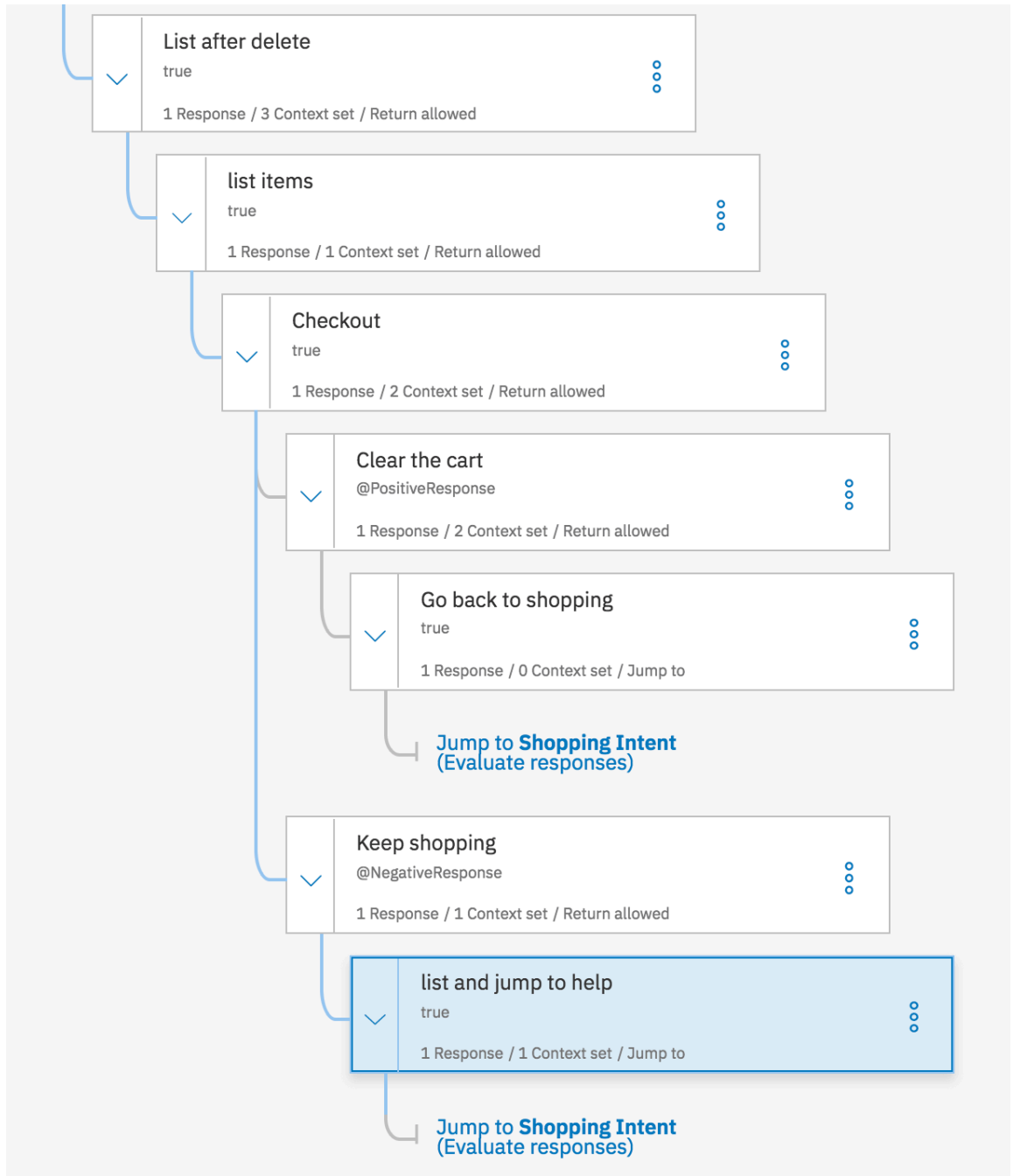


55. Add a “Jump to Shopping Intent” node after the “list and jump to help node.

56. Click the menu on the “list and jump to help” node and then click Jump to.

57. Scroll to the beginning of the conversation, click the “Shopping Intent” node, and then click Respond.

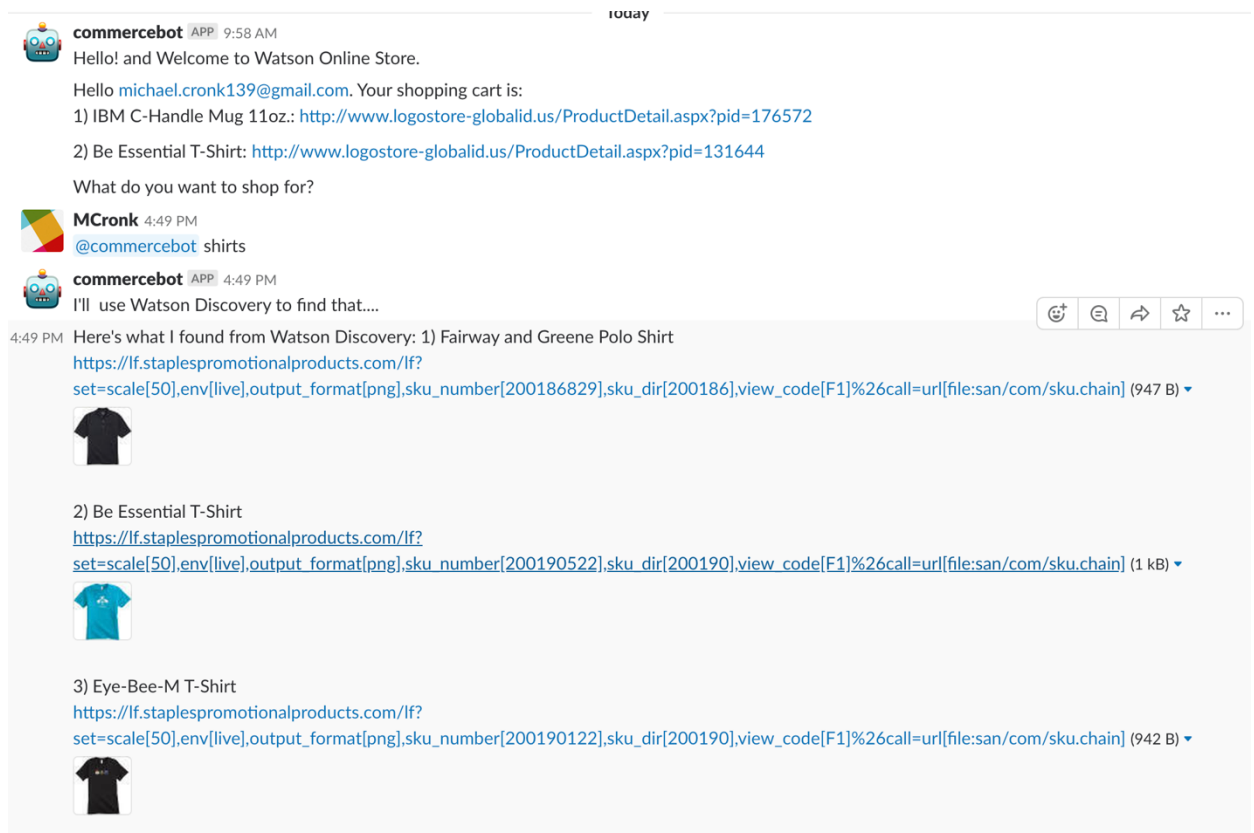
58. Make sure the conversation nodes you added and modified look like the following figure:



Lesson 1.7 Update the Python code to add the checkout function and redeploy the app.

Pre-requisites:

- Completion of Lessons 1.1 – 1.6.
1. Return to the watson-online-store Slack channel and verify that your bot is still active. If the bot is not active, repeat lesson 1.3 to verify that your environment variables are still set to the correct values.
 2. Talk with your chatbot. Try the checkout function after you have added an item to your cart and deleted an item.




The screenshot shows a Slack message history in a channel named 'louay'. The messages are as follows:


- commercebot** (APP) 9:58 AM: Hello! and Welcome to Watson Online Store. Hello michael.cronk139@gmail.com. Your shopping cart is:
1) IBM C-Handle Mug 11oz.: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=176572>
2) Be Essential T-Shirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=131644>
What do you want to shop for?
- MCronk** 4:49 PM: @commercebot shirts
- commercebot** (APP) 4:49 PM: I'll use Watson Discovery to find that...

A message from 4:49 PM follows, containing search results from Watson Discovery:


Here's what I found from Watson Discovery: 1) Fairway and Greene Polo Shirt
[https://lf.staplespromotionalproducts.com/lf?set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200186829\],sku_dir\[200186\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200186829],sku_dir[200186],view_code[F1]%26call=url[file:san/com/sku.chain]) (947 B) ▾



2) Be Essential T-Shirt
[https://lf.staplespromotionalproducts.com/lf?set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200190522\],sku_dir\[200190\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200190522],sku_dir[200190],view_code[F1]%26call=url[file:san/com/sku.chain]) (1 kB) ▾



3) Eye-Bee-M T-Shirt
[https://lf.staplespromotionalproducts.com/lf?set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200190122\],sku_dir\[200190\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200190122],sku_dir[200190],view_code[F1]%26call=url[file:san/com/sku.chain]) (942 B) ▾



© Copyright IBM Corp. 2017

Course materials may not be reproduced in whole or in part without the prior written permission of IBM


5) Applique Crew Sweatshirt


[https://lf.staplespromotionalproducts.com/lf?](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200275582],sku_dir[200275],view_code[F1]%26call=url[file:san/com/sku.chain])

[set=scale\[50\],env\[live\],output_format\[png\],sku_number\[200275582\],sku_dir\[200275\],view_code\[F1\]%26call=url\[file:san/com/sku.chain\]](https://lf.staplespromotionalproducts.com/lf?set=scale[50],env[live],output_format[png],sku_number[200275582],sku_dir[200275],view_code[F1]%26call=url[file:san/com/sku.chain]) (1 kB) ▾



Please choose which item you will add to your cart.

 **MCronk** 4:49 PM
[@commercebot](#) 5

 **commercebot** APP 4:49 PM
Let's add to your shopping cart....
OK, let's list your shopping cart...

Your cart is:


- 1) IBM C-Handle Mug 11oz.: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=176572>
- 2) Be Essential T-Shirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=131644>
- 3) Applique Crew Sweatshirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=206347>



4:49 PM Which item number do you want to delete?

 **MCronk** 4:49 PM
[@commercebot](#) 3

 **MCronk** 4:49 PM
[@commercebot](#) 3

 **commercebot** APP 4:49 PM
Let's Delete 3.


Now we'll list the cart...

Items in your cart are now:

- 1) IBM C-Handle Mug 11oz.: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=176572>
- 2) Be Essential T-Shirt: <http://www.logostore-globalid.us/ProductDetail.aspx?pid=131644>

Would you like to checkout and clear the cart?

 **MCronk** 4:50 PM
[@commercebot](#) yes

 **commercebot** APP 4:50 PM
Ok, let's clear your cart.

Shopping cart has been cleared!

Hello michael.cronk139@gmail.com. Your shopping cart is:

checkout

What do you want to shop for?



You have completed the lab.