

IBM Cloud

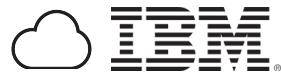
Build a Cognitive Assistant

Using Watson Conversation



Lab Guide





Notices and Disclaimers

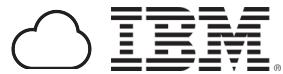
© Copyright IBM Corporation 2018.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others



Document Revision History

Rev #	File Name	Date
.5	NLP and STT Lab Guide	1/17/18
1.0	Build a Cognitive Assistant – Lab Guide	1/22/2018

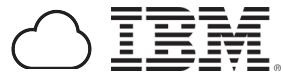
Prepared & Revised by:

Chris Harrold – chris.harrold@ibm.com



Table of Contents

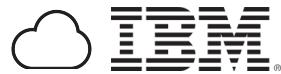
Lab Environment Overview	5
Module 1: Deploying NodeRed Starter	6
Module 1: Lab Instructions	7
Module 1: Lab Summary	14
Module 2: Deploy Watson Services	15
Module 2: Lab Instructions	16
Module 2: Lab Summary	22
Module 3: Train Watson Conversation	23
Module 3: Lab Instructions	24
Module 3: Lab Summary	35
Module 4: Build the Node-RED Flow	36
Module 4: Lab Instructions	37
Module 4: Lab Summary	53



Lab Environment Overview

Installed Software and Tools

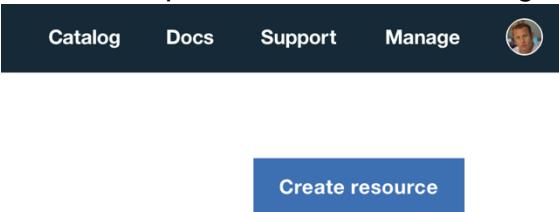
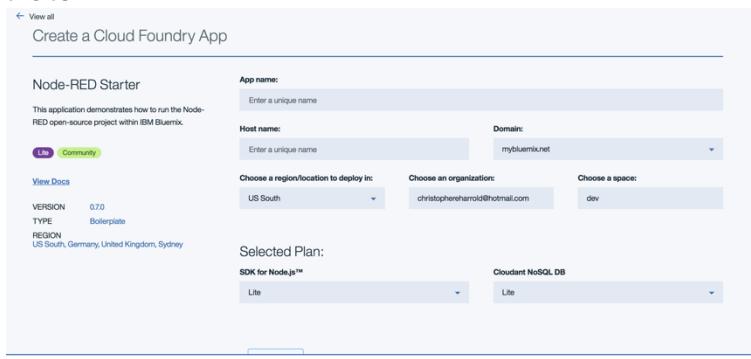
Software	Link
None – all IBM Cloud Based	Console.bluemix.net

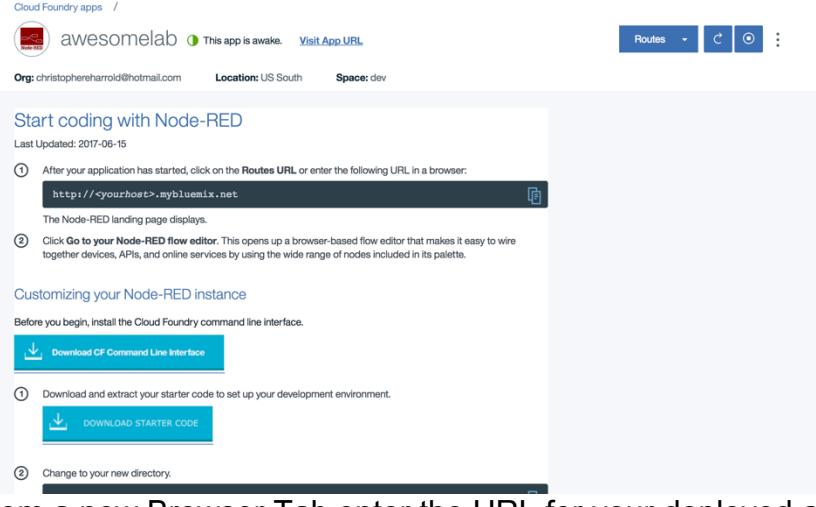
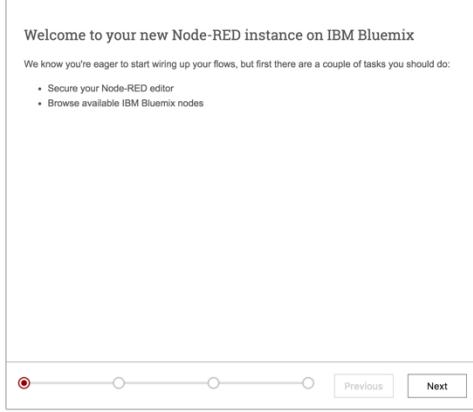


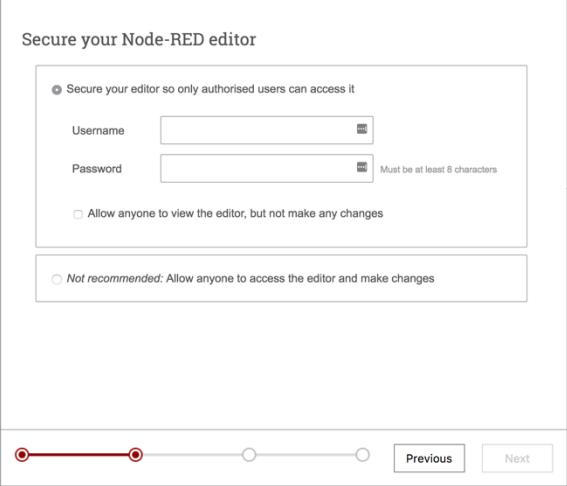
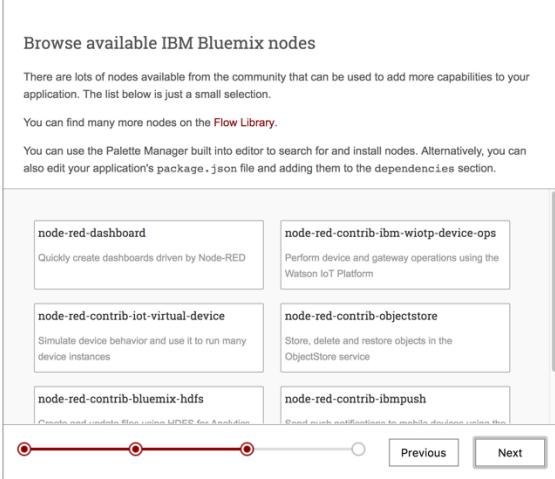
Module 1: Deploying NodeRED Starter

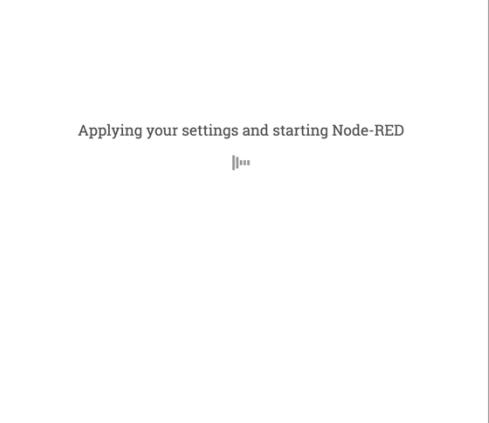
Purpose:	This lab introduces the subject of deploying the NodeRED starter app. After completing the lab, you should be able to: <ul style="list-style-type: none">• Access the Console, have a secured editor instance for NodeRED flows, and be ready to build your flow.
Tasks:	Tasks you will complete in this lab exercise include: <ul style="list-style-type: none">○ Deploy NodeRED Starter○ Secure the Editor○ Log In

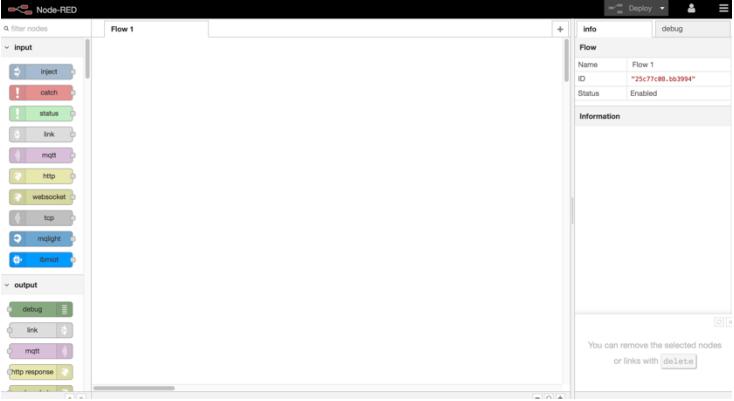
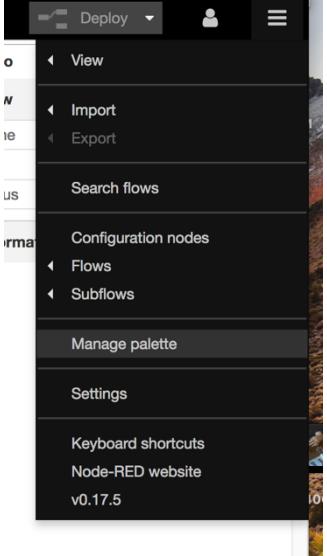
Module 1: Lab Instructions

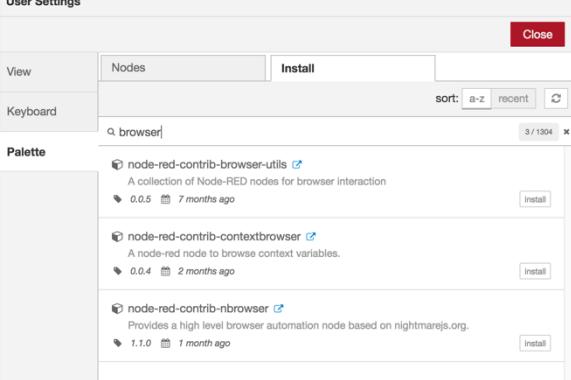
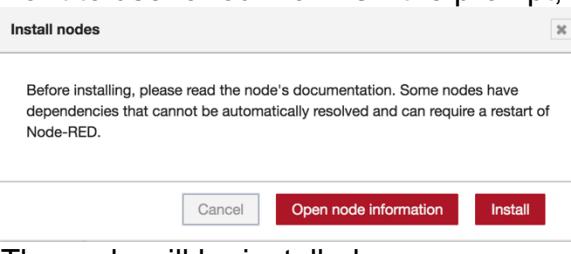
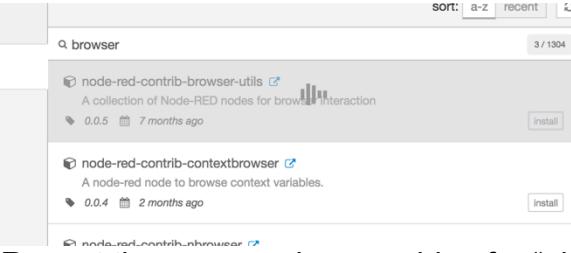
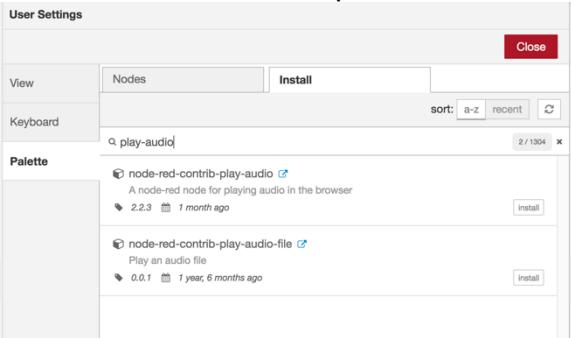
Step	Action
1	<p><u>Log Into IBM Cloud Account</u></p> <ol style="list-style-type: none"> Log in to your IBM Cloud Account – https://console.bluemix.net From the top menu, click on: “Catalog” 
2	<p><u>Deploy NodeRED Starter</u></p> <ol style="list-style-type: none"> From the Catalog Page, browse to or search for “Node-RED” Select the Node-RED Starter app (labeled with a purple “Lite”) You will arrive at the deploy screen 
3	<p><u>Configure the Node-RED App</u></p> <ol style="list-style-type: none"> Fill in an Application name – write down your appname so you can access it later: The hostname will populate with your appname. The URL to access it is: 

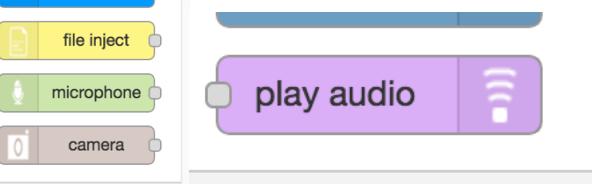
Step	Action
	<p>{YourAppName}.mybluemix.net</p> <p>c. Click the “Create” button in the lower right</p>
4	<p>Access and Customize Service</p> <p>a. Your service will begin deploying as soon as you finish the previous step</p> <p>b. You will see the service status change to “awake” indicating the service is deployed:</p>  <p>c. From a new Browser Tab enter the URL for your deployed app as shown in the previous step</p> <p>d. You will be greeted with the Node-RED welcome screen:</p>  <p>e. Click Next to customize your instance</p> <p>f. Supply a username and password to access the editor: (for the purposes of this lab you can simply say “Allow anyone to access and make changes” but this is NOT a production option and should be used cautiously!)</p>

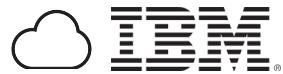
Step	Action
	 <p data-bbox="365 789 1459 861">g. Browse available nodes – feel free to look, but we will be customizing the Node-RED instance later – click the “Next” button to continue</p>  <p data-bbox="365 1360 1475 1431">h. Finish The Install screen – your options will be displayed and you can click “Finish”:</p>

Step	Action
	<p>Finish the install</p> <p>You have made the following selections:</p> <ul style="list-style-type: none"> Secure your editor so only authorised users can access it <p>You can change these settings at any time by setting the following environment variables via the Bluemix console:</p> <ul style="list-style-type: none"> NODE_RED_USERNAME - the username NODE_RED_PASSWORD - the password NODE_RED_GUEST_ACCESS - if set to 'true', allows anyone read-only access to the editor <p style="text-align: center;">  Previous Finish </p> <p>i. Your settings will be applied and you will be taken to the Welcome page when the configuration completes:</p> <p style="text-align: center;">  </p> <p style="text-align: center;">  </p> <p>Node-RED Flow-based programming for the Internet of Things</p> <p>Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.</p> <p>This instance is running as an IBM Bluemix application, giving it access to the wide range of services available on the platform.</p> <p>More information about Node-RED, including documentation, can be found at nodered.org.</p> <p>Go to your Node-RED flow editor</p> <p>Learn how to customise Node-RED</p> <p>Customising your instance of Node-RED</p> <p>This instance of Node-RED is enough to get you started creating flows.</p> <p>You may want to customise it for your needs, for example replacing this introduction page with your own, adding http authentication to the flow editor or adding new nodes to the palette.</p> <p>j. Click on the “Go to your Node-RED flow editor”</p>

Step	Action
	 <p>k. We can now create our flow for the lab!</p>
5	<p><u>Customize our Pallette</u></p> <p>a. In the flow editor screen, select “Manage Palette” from the menu to open the palette editor:</p>  <p>b. The palette manager will open – click on the “install” tab:</p>

Step	Action
	 <p>c. In the search field type in “browser” – next to the first result returned, click the small “Install” button – this will install some additional Nodes that we will want to use for our flow. On the prompt, click “Install”:</p>  <p>d. The node will be installed:</p>  <p>e. Repeat the process, by searching for “play-audio” and selecting the “Install” button next to the first option:</p>  <p>f. After completion, click the “Close” Button in the upper right. If you look in the node list on the left, you should now see several new nodes that we will use in our flow for this lab:</p>

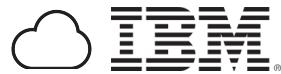
Step	Action
	 <p>g. You have now completed all the setup required for Node-RED. In the next part of the lab, we will deploy the services we need for our Speech-to-Text and NLP.</p>



Module 1: Lab Summary

In the first lab we deployed and customized the Node-RED starter app. We now have access to a secured editing environment for our application flow. We added the libraries we need to ensure that we can create the flow and access all the various resources needed to execute the flow we are going to build.

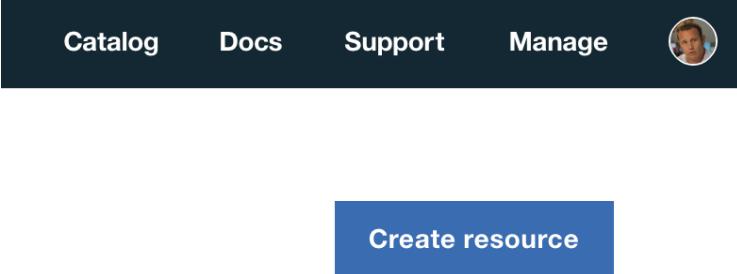
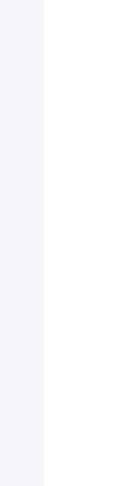
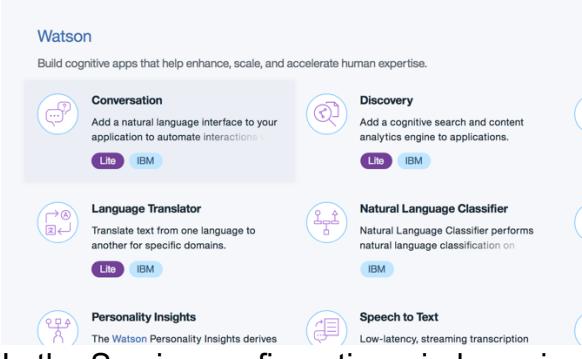
At this point we will move on to Lab 2 and deploy the Watson Services we will consume from our Node-RED application flow.

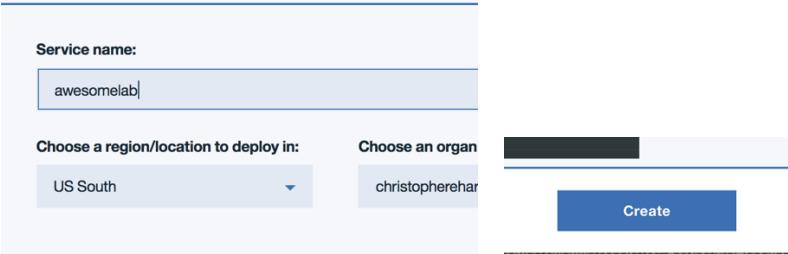
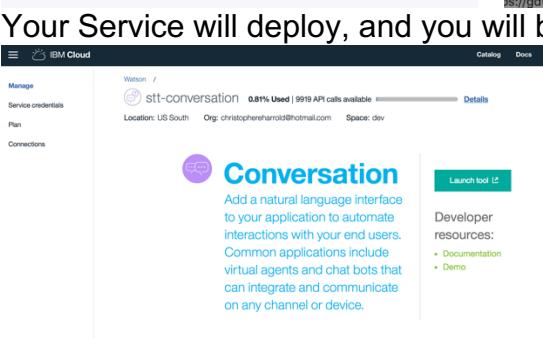
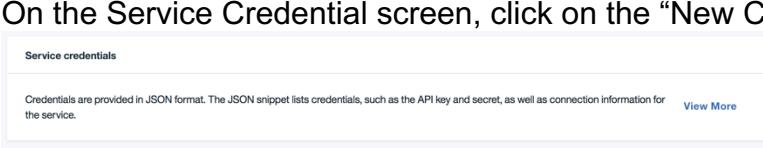


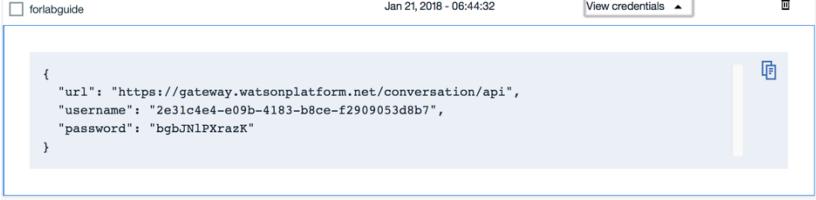
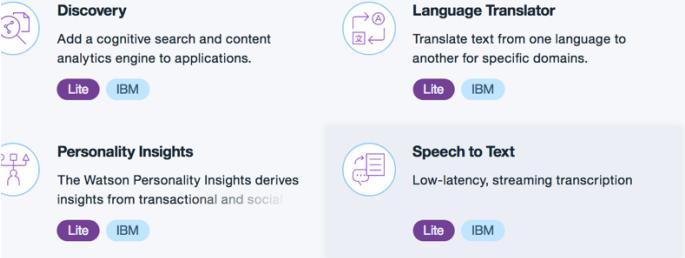
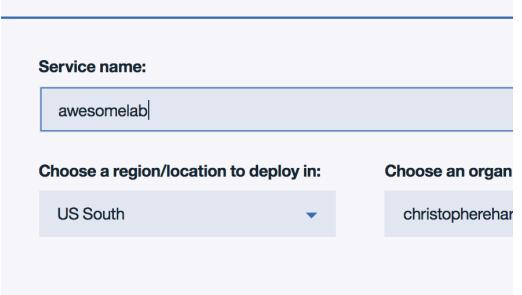
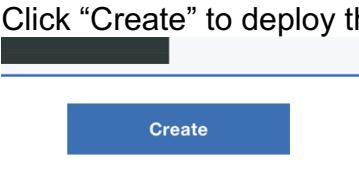
Module 2: Deploy Watson Services

Purpose:	<p>This lab introduces the subject of deploying Watson services on IBM Cloud. After completing the lab, you should be able to:</p> <ul style="list-style-type: none">• Choose Watson services from the catalog• Customize and configure services for deployment• Identify connection credentials and API endpoints for our application flow
Tasks:	<p>Tasks you will complete in this lab exercise include:</p> <ul style="list-style-type: none">• Deploy Watson Conversation service• Deploy Watson Speech-To-Text• Deploy Watson Text-to-Speech• Identify and record connection details for each service

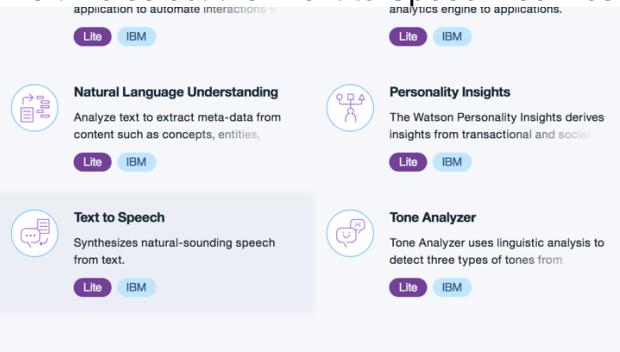
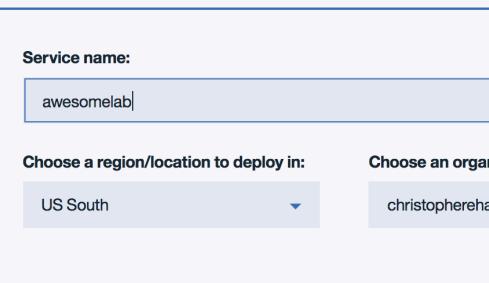
Module 2: Lab Instructions

Step	Action
1	<p><u>Deploy Watson Conversation Services</u></p> <p>a. From the top menu in your IBM Cloud Account, select “Catalog”</p>  <p>b. From the left-hand menu, scroll down and click on “Watson”</p>  <p>c. In the services catalog window, choose “Conversation” by clicking on it</p>  <p>d. In the Service configuration window, give your conversation service a name (it can be anything) and click the big “Create” Button at bottom right</p>

Step	Action
	<p>e. Your Service will deploy, and you will be taken to the service screen</p>  <p>e. Your Service will deploy, and you will be taken to the service screen</p>  <p>f. Before doing anything else, select “Service Credentials” from the left-hand menu</p>  <p>g. On the Service Credential screen, click on the “New Credential” button</p>  <p>h.</p> <p>i. Give your credentials a name, or take the default on the pop-up screen, and click “Add”</p> <p>j. When you are returned to the Service Credential menu, you should now see a “KEY NAME” listed under the button you clicked to create the credentials</p>

Step	Action
	 <p>k. Click on the “View Credentials” next to the key pair and open the credentials panel for your created credentials l. The credentials will appear below the key pair. Make a copy of this information (into notepad/wordpad/etc...) for later use in the Node-RED flow (You can always come back and get it later, but this saves you poking around during the rest of the lab)</p>
2	<p><u>Deploy Speech to Text</u></p> <p>a. From the top menu, click “Catalog” again to bring up the service catalog b. Like last time, we want to scroll down on the left and click on “Watson” c. This time select the “Speech to Text” service by clicking on it</p>  <p>d. The service configuration screen will load. Give the service a name – any name will work, just make it descriptive for you</p>  <p>e. Click “Create” to deploy the service</p>  <p>f. The service will deploy, and take you to the Speech to Text Service menu</p>

Step	Action
	<p>Watson / awesomelab-stt Location: US South Org: christophereharrold@hotmail.com Space: dev</p> <p> Speech to Text Low-latency, streaming transcription</p> <p>Developer resources:</p> <ul style="list-style-type: none"> • Getting started tutorial • Demo <p>g. As we did with the last service, select “Service Credentials” from the left-hand menu</p> <p>Getting started Manage Service credentials Plan Connections</p> <p>h. On the Service Credential screen, check if there is an existing credential Key Pair. If there is, skip to step “g”. If not click on the “New Credential” button and follow these steps:</p> <ul style="list-style-type: none"> • Give your credentials a name, or take the default on the pop-up screen, and click “Add” • When you are returned to the Service Credential menu, you should now see a “KEY NAME” listed under the button you clicked to create the credentials <p>i. Click on the “View Credentials” next to the key pair and open the credentials panel for your created credentials</p> <p></p> <p>j. The credentials will appear below the key pair. Make a copy of this information (into notepad/wordpad/etc...) for later use in the Node-RED flow (You can always come back and get it later)</p>
3	<p><u>Deploy Text to Speech</u></p> <p>a. From the top menu, click “Catalog” again to bring up the service catalog</p> <p>b. Like last time, we want to scroll down on the left and click on “Watson”</p>

Step	Action
	<p>c. This time select the “Text to Speech” service by clicking on it</p>  <p>d. The service configuration screen will load. Give the service a name – any name will work, just make it descriptive for you</p>  <p>e. Click “Create” to deploy the service</p>  <p>f. The service will deploy and take you to the Text to Speech Service menu</p>  <p>g. As we did with the last service, select “Service Credentials” from the left-hand menu</p>

Step	Action
	<p>Getting started Manage Service credentials Plan Connections</p> <p>h. On the Service Credential screen, check if there is an existing credential Key Pair. If there is, skip to step "g". If not click on the "New Credential" button and follow these steps:</p> <ul style="list-style-type: none"> • Give your credentials a name, or take the default on the pop-up screen, and click "Add" • When you are returned to the Service Credential menu, you should now see a "KEY NAME" listed under the button you clicked to create the credentials <p>i. Click on the "View Credentials" next to the key pair and open the credentials panel for your created credentials</p>  <pre> { "url": "https://gateway.watsonplatform.net/conversation/api", "username": "2e31c4e4-e09b-4183-b8ce-f2909053d8b7", "password": "bgbJNlPXrazK" } </pre> <p>j. The credentials will appear below the key pair. Make a copy of this information (into notepad/wordpad/etc...) for later use in the Node-RED flow (You can always come back and get it later)</p>



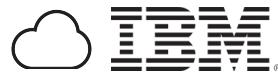
Module 2: Lab Summary

You have now deployed the services required to complete the lab:

- Watson Conversation
- Watson Speech to Text
- Watson Text to Speech

Well done! Now we will go through the process of training the Conversation Service so that we can have an interaction with it through our assistant.

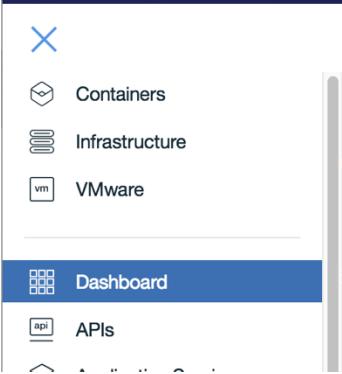
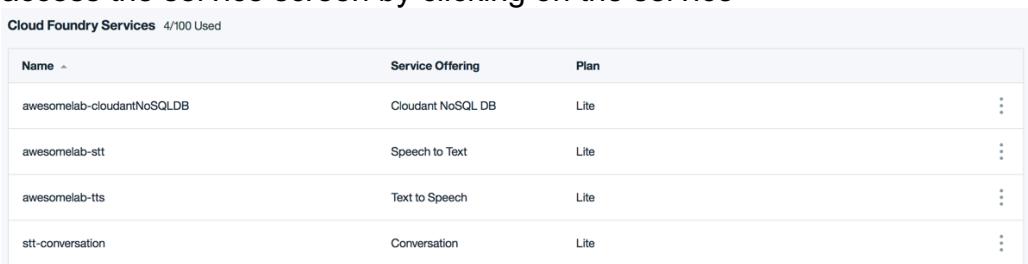
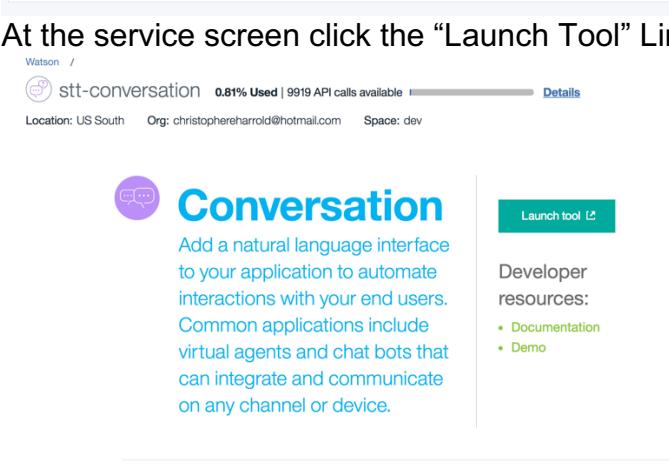
Proceed to lab 3 to see how to train your Watson Conversation Agent,

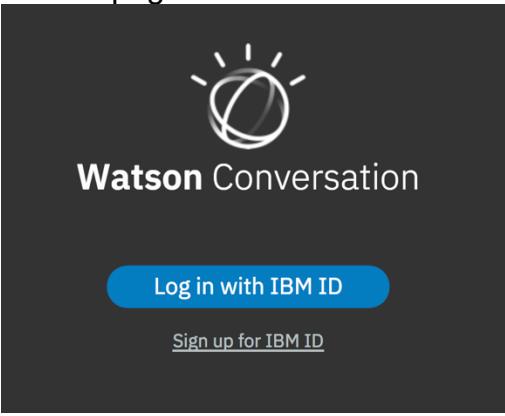
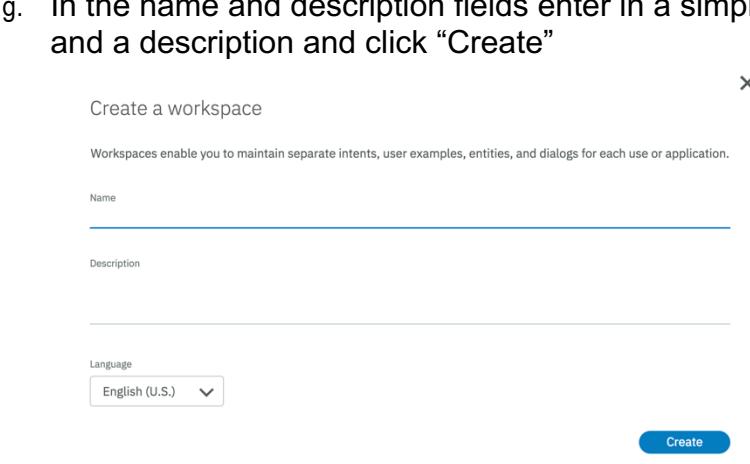


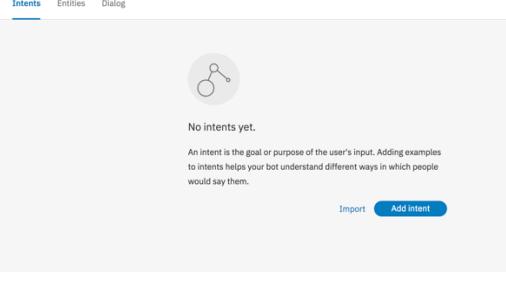
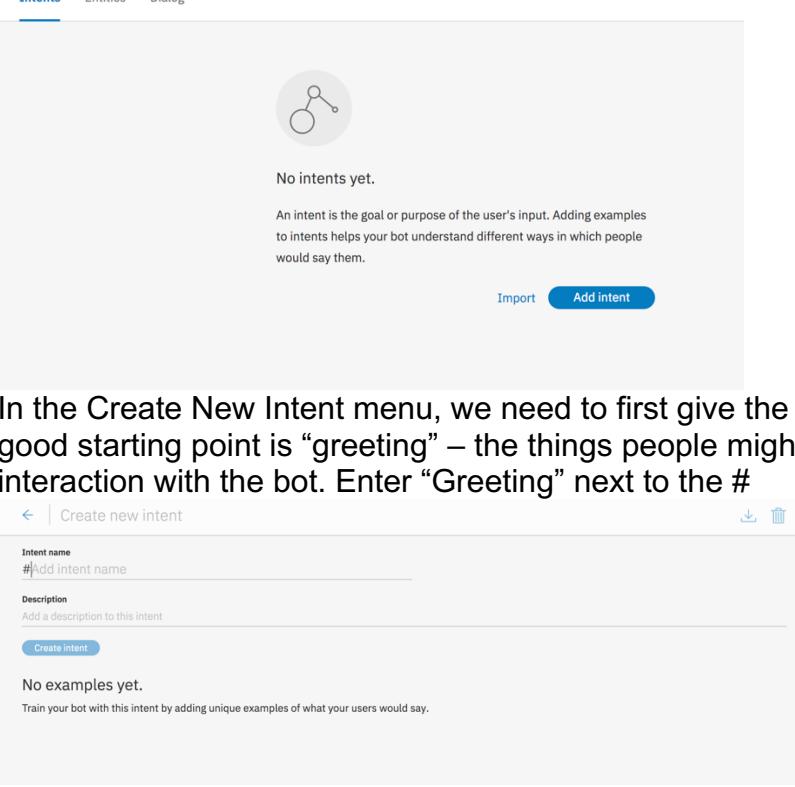
Module 3: Train Watson Conversation

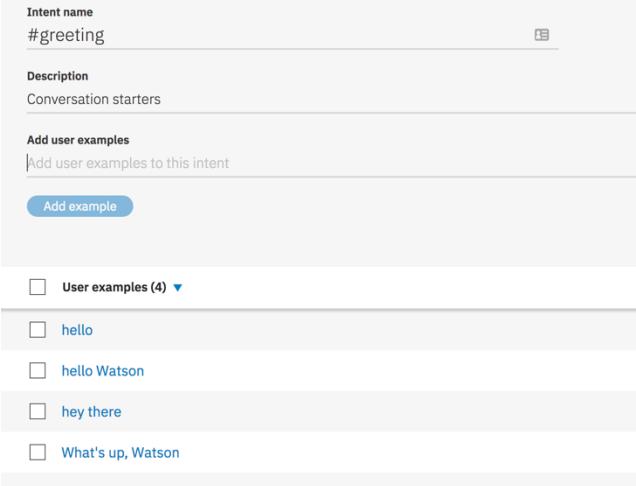
Purpose:	<p>This lab will show you the basics of training a Watson Conversation Agent to have a basic conversation. It will also touch on ways that it could be expanded for deeper exploration on your own. When finished with this lab, you will:</p> <ul style="list-style-type: none">• Understand Intents, Entities, and Dialog options as they relate to Watson Conversation• Create basic Intents• Create a basic Dialog path• Understand how to expand Watson Conversation beyond the basic lab
Tasks:	<p>Tasks you will complete in this lab exercise include:</p> <ul style="list-style-type: none">• Create Intents<ul style="list-style-type: none">◦ Hello◦ Goodbye◦ Questions• Create Dialog flow(s)• Train Watson Conversation on the fly• Interact with your created Intents and Dialogs

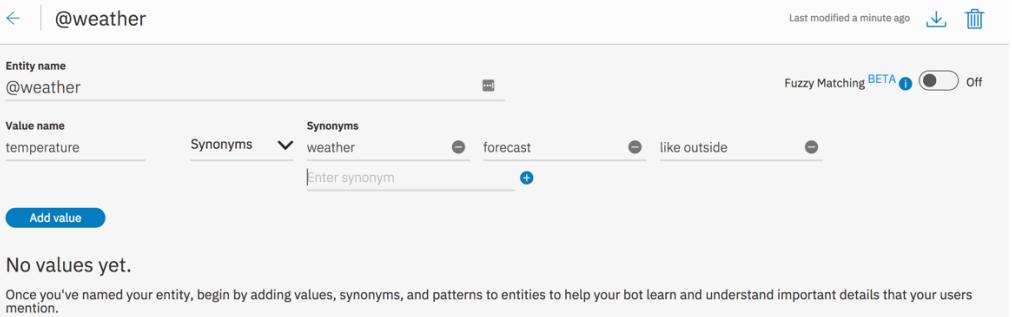
Module 3: Lab Instructions

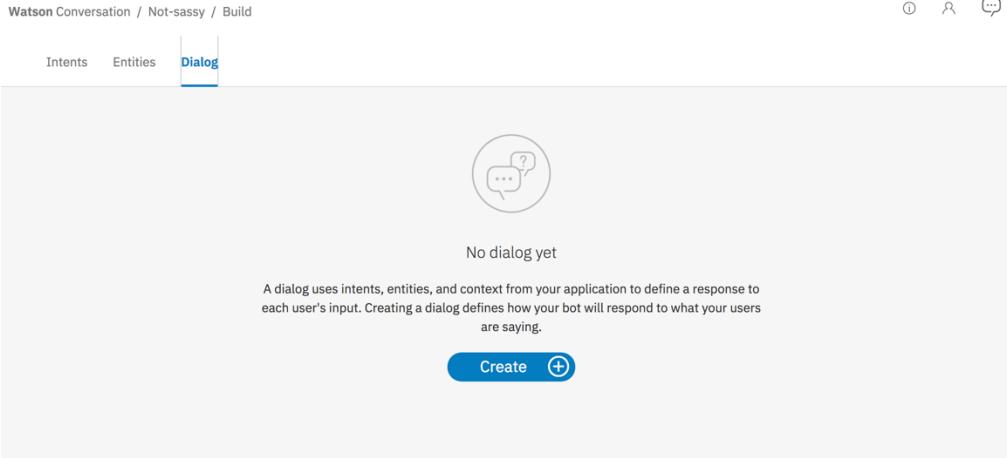
Step	Action
1	<p><u>Access the Watson Conversation Trainer</u></p> <p>a. The Watson Conversation service has a very easy to use trainer for conversations built-in to the service. It provides a very simple framework for building conversations. From your IBM Cloud Menu, select the top left “Hamburger” menu and choose “Dashboard”</p>  <p>b. From the Dashboard view, find your Conversation Service in the list, and access the service screen by clicking on the service</p>  <p>c. At the service screen click the “Launch Tool” Link</p> 

Step	Action
	<p>d. You will be taken to the Watson Conversation login page – click “Log In with Your IBM ID” and you will be taken to the Watson Conversation Trainer page</p>  <p>e. When it opens, you will see that there is already a demonstration Workspace created. Workspaces are what holds all the parts of your conversations.</p> <p>f. For the lab, we will want to create a new workspace, so click on “Create” at the top</p>  <p>g. In the name and description fields enter in a simple name for our lab-bot and a description and click “Create”</p>  <p>h. You will be taken to the Workspace, which is blank, and greeted with the default workspace screen</p>

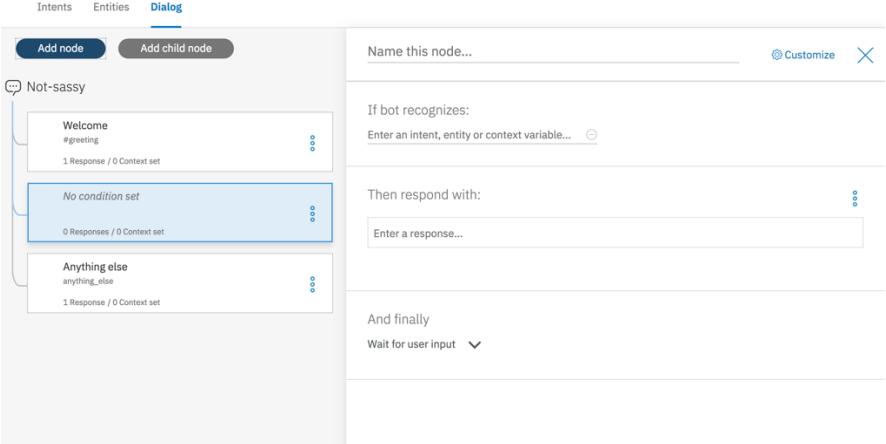
Step	Action
	
2	<p><u>Intents</u></p> <p>a. Intents are the basics of conversations – they tell your bot how to react to what the user says. When you create your Workspace above, the system will prompt you to add or import Intents. Importing intents is very useful if you have a “real” conversation you want to import, since there can be many hundreds of Intents in a deep conversation. For the purposes of the lab, we will simply click, “Add Intent”</p>  <p>b. In the Create New Intent menu, we need to first give the intent a name. A good starting point is “greeting” – the things people might say to start an interaction with the bot. Enter “Greeting” next to the #</p> <p>c. In the description field we can provide information about what we meant with this Intent (extremely useful if you are working with others!). Add a simple description in the field</p>

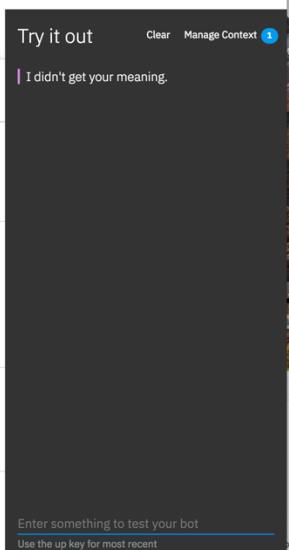
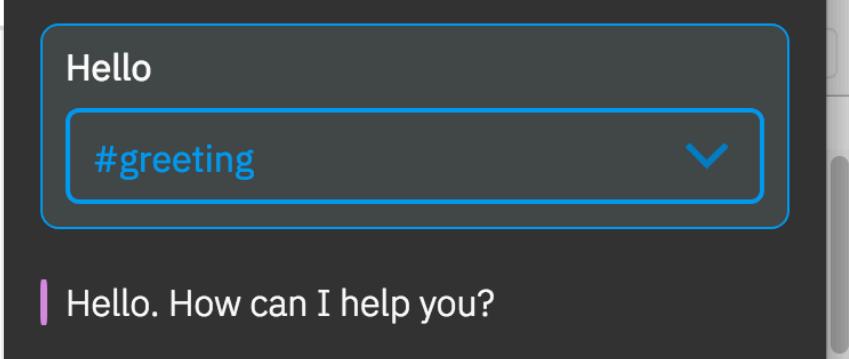
Step	Action
	<p>d. When done, click “Create Intent”</p> <p>e. You will see a green floating confirmation that says “#greeting added. Notice now too, there is a new line that says “Add user examples”. This is where we enter our text that we think a user might say:</p> <ul style="list-style-type: none"> • This can be a single word, like “Hello”, “Goobye”, “Yes”, “No” etc... • It can also be a short phrase, like “Hello, Watson”, “See you later”, “That’s not right”, etc... <p>f. Enter a few examples that match the Intent, by typing them in, and clicking “Add Example” – 5 or 6 is plenty for the lab</p>  <p>g. When you are satisfied with the list, click on the blue arrow next to the Intent name at the very top to go back to the main menu</p> <p>h. You will now see your #greeting Intent in the list</p> <p>i. Now let's create the remaining Intents for our lab:</p> <ul style="list-style-type: none"> • Click “Add intent”, give it a name (Let's use “Goodbye” and “Question” for the other two intents) • Add some examples as you did in the first one. • For “Question” you can use phrases like: <ul style="list-style-type: none"> • What time is • How do I • Where is • When is • Show me how to <p>j. When you are done, your list should look like the one below:</p>

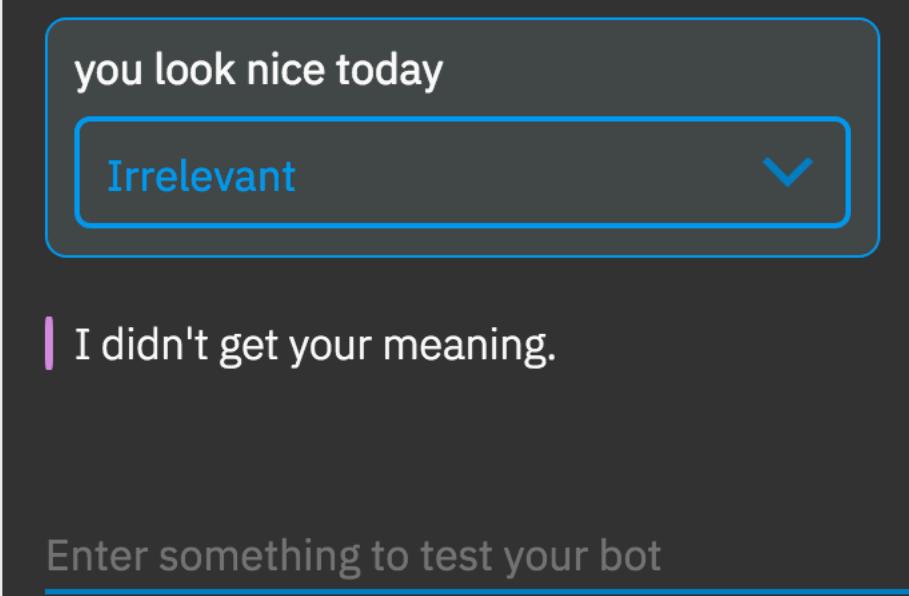
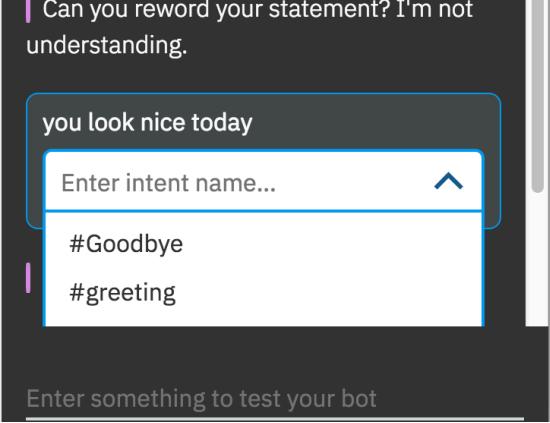
Step	Action																
	<p>Watson Conversation / Not-sassy / Build</p> <p>Intents Entities Dialog</p> <p>Add intent</p> <table border="1"> <thead> <tr> <th>Intent (3) ▾</th> <th>Description</th> <th>Modified</th> <th>Examples</th> </tr> </thead> <tbody> <tr> <td>#Goodbye</td> <td></td> <td>a few seconds ago</td> <td>1</td> </tr> <tr> <td>#greeting</td> <td>Conversation starters</td> <td>4 minutes ago</td> <td>4</td> </tr> <tr> <td>#question</td> <td></td> <td>a few seconds ago</td> <td>2</td> </tr> </tbody> </table> <p>k. Well done! Now that we know what a user might say, it's time we figure out how to respond.</p>	Intent (3) ▾	Description	Modified	Examples	#Goodbye		a few seconds ago	1	#greeting	Conversation starters	4 minutes ago	4	#question		a few seconds ago	2
Intent (3) ▾	Description	Modified	Examples														
#Goodbye		a few seconds ago	1														
#greeting	Conversation starters	4 minutes ago	4														
#question		a few seconds ago	2														
3	<p>Entities</p> <p>a. We are not going to use entities in the lab very much, but it is worth mentioning them as we are here, because they can be very important to a full production project with conversation. Entities provide a means of creating patterns or synonyms for words a user might say. In this way it helps pre-train the response so that you don't have to fully train it on the fly all the time.</p> <p>b. Let's create a quick entity so you can see how it works. Click the "Entities" menu from the menu bar</p> <p>c. Again, you are prompted to import or add, we will click "Add Entity"</p> <p>d. In the new entity screen, you will first give it a name – let's use "weather" and click "Create entity"</p> <p></p> <p>e. You are now prompted to enter the root phrase you want to target and either synonyms or patterns. For spoken-word bots, synonyms will generally work best, so let's add one now. Type in "weather" in the "Value Name" field, and hit the "Tab" key</p> <p>f. For synonyms, this could be a situation where your bot will give someone the temperature outside. This is where you should think of possible ways someone could ask, "what's the weather?" For synonyms, a very short list could be: "temperature", "what's it like outside", "is it hot" etc. Enter a</p>																

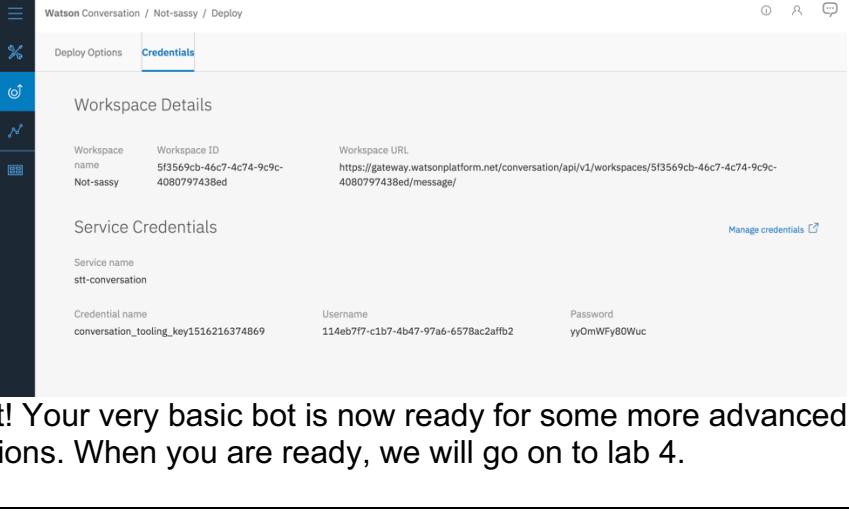
Step	Action
	<p>synonym, and then click the “Add Value” button to add the new value to the list</p> <p>g. Once done, you can go back by clicking the blue arrow at the top h. You will now see your entity in the list</p>
4	<p><u>Dialog</u></p> <p>a. Dialog is the “meat” of the Watson Conversation interface. It is what defines a conversation flow, how the system interacts with the user, and the path(s) that they take through the system. For our lab, we will be building a very simple conversation dialog tree. Click on “Dialog” to open the dialog creation window.</p> <p>b. Click the big blue “Create” button in the middle of the screen.</p>  <p>c. The system will create two basic place holder dialog nodes in the workspace:</p> <ul style="list-style-type: none"> • Welcome – this is a generic “welcome to the bot” dialog option • Anything_else – this is the “catch all” in case something the bot does not recognize comes in, it allows the bot to respond with a clarification question such as “I did not understand, can you rephrase?” <p>d. Click on the “Welcome” dialog node and the options window will open</p>

Step	Action
	<p>Welcome  </p> <p>If bot recognizes: <input type="text" value="Enter an intent, entity or context variable..."/> </p> <p>Filter by:</p> <ul style="list-style-type: none"> <input type="checkbox"/> # intents <input type="checkbox"/> @ entities <input type="checkbox"/> \$ context variables <p>Common conditions:</p> <ul style="list-style-type: none"> welcome Triggers when a conversation is started by the system anything_else Triggers when the user input does not match <p>And finally <input type="text" value="Wait for user input"/> </p> <p>e. Click on the “-“ symbol next to the word Welcome to remove this option – we will replace it with our own text</p> <p>f. Click into the now empty field and a context menu will appear (shown above)</p> <p>g. We want to use the Intent we created earlier, so click on the “# intents” option, and select the “#greeting” intent from the list</p> <p>Welcome  </p> <p>If bot recognizes: <input type="text" value="#greeting"/>  </p> <p>Then respond with:  <ul style="list-style-type: none"> 1. Hello. How can I help you?  2. greetings!  3. hello  <input type="text" value="Add a variation to this response"/> Variations are sequential. Set to random </p> <p>h. The second step is to tell the bot what to “say” in return – A default is already present, but you can add other variations to the list <ul style="list-style-type: none"> • Add additional potential responses, by typing them into the field and hitting “Enter” </p> <p>i. Once you have added one or more alternatives, it is possible to change the way the bot will respond by choosing either the “Set to sequential” or “Set to random” link below your list of options. This can help with making the bot seem more realistic (giving a different response makes it feel less scripted)</p>

Step	Action
	<p>and more natural).</p> <p>j. Once you have added your alternative, and decided on the order, simply click the blue “X”</p>  <p>k. Now let's create a new dialog node, by clicking the “Add node” button at the top of the dialog configuration page</p> <p>l. A new Node will appear with the text “No Condition set” and the configuration menu will pop up on the right.</p>  <p>m. Just like we did before, we give it a name in the Name field, this time let's use “Goodbye”</p> <p>n. In the next field, click the “-“ sign and choose the “# intents” option, and then choose the “#goodbye” intent we created earlier</p> <p>o. Now as before, add in the possible answers to signing off from your bot, and change the order (if desired)</p> <p>p. Repeat steps “K” through “O” to create a “Question” node</p>
5	<p>Testing and Training</p> <p>a. Now that our Bot knows what to say and why to say it, we can quickly test it right from the Watson Conversation builder. Click on the “Ask Watson” button on the top right of the screen (it looks like cartoon dialog bubble).</p> 

Step	Action
	<p>b. The conversation window will open</p>  <p>c. In the bottom field you can enter in text that will prompt a response from the bot – start with “Hello”</p> <p>d. The bot will respond based on the detected word or phrase and show you what Intent it is responding with – in this case, the #greeting intent</p>  <p>e. Now let's see what happens when Watson doesn't quite know what we are saying – enter “You look nice today” in the field – you should get the response shown in the image below:</p>

Step	Action
	 <p data-bbox="372 876 1490 1072">f. Watson didn't understand! That's OK, because you can train Watson directly from this tool. In the response field, click the drop-down button next to the detected Intent (Irrelevant in this case). You will see that you can assign an intent to the phrase, and it will be used to "train" Watson to recognize that phrase as whatever Intent you specify</p>  <p data-bbox="372 1495 1473 1564">g. Try and few more questions, greetings, and sign-off phrases, until you are satisfied that your bot is responding the way you expected.</p> <p data-bbox="372 1564 1473 1784">h. Our last step in the lab is to get ready to interact with our bot, using Node-RED. To do this we need to collect a couple of pieces of information:</p> <ul style="list-style-type: none"> <li data-bbox="474 1649 1465 1691">• Click the "Deploy" Button on the left side menu (It looks a bit like a fish-hook with a circle in it – see screenshot below) <li data-bbox="474 1691 1171 1723">• From the deploy options, select "Credentials" <li data-bbox="474 1723 1481 1755">• Record the Workspace ID, Username, and Password from this page

Step	Action															
	 <p>Watson Conversation / Not-sassy / Deploy</p> <p>Deploy Options Credentials</p> <p>Workspace Details</p> <table border="1"> <tr> <td>Workspace name</td> <td>5f3569cb-46c7-4c74-9c9c-4080797438ed</td> <td>Workspace ID</td> <td>5f3569cb-46c7-4c74-9c9c-4080797438ed</td> <td>Workspace URL</td> <td>https://gateway.watsonplatform.net/conversation/api/v1/workspaces/5f3569cb-46c7-4c74-9c9c-4080797438ed/message/</td> </tr> </table> <p>Service Credentials</p> <table border="1"> <tr> <td>Service name</td> <td>srt-conversation</td> <td>Manage credentials</td> </tr> <tr> <td>Credential name</td> <td>conversation_tooling_key1516216374869</td> <td>Username</td> <td>114eb7f7-c1b7-4b47-97a6-6578ac2affb2</td> <td>Password</td> <td>yyOmWFy80Wuc</td> </tr> </table> <p>i. That's it! Your very basic bot is now ready for some more advanced interactions. When you are ready, we will go on to lab 4.</p>	Workspace name	5f3569cb-46c7-4c74-9c9c-4080797438ed	Workspace ID	5f3569cb-46c7-4c74-9c9c-4080797438ed	Workspace URL	https://gateway.watsonplatform.net/conversation/api/v1/workspaces/5f3569cb-46c7-4c74-9c9c-4080797438ed/message/	Service name	srt-conversation	Manage credentials	Credential name	conversation_tooling_key1516216374869	Username	114eb7f7-c1b7-4b47-97a6-6578ac2affb2	Password	yyOmWFy80Wuc
Workspace name	5f3569cb-46c7-4c74-9c9c-4080797438ed	Workspace ID	5f3569cb-46c7-4c74-9c9c-4080797438ed	Workspace URL	https://gateway.watsonplatform.net/conversation/api/v1/workspaces/5f3569cb-46c7-4c74-9c9c-4080797438ed/message/											
Service name	srt-conversation	Manage credentials														
Credential name	conversation_tooling_key1516216374869	Username	114eb7f7-c1b7-4b47-97a6-6578ac2affb2	Password	yyOmWFy80Wuc											



Module 3: Lab Summary

Congratulations, your bot is created and trained. At this point you have a functioning chat-bot that can be interacted with in several ways. It is simple, but it provides the foundation for understanding and developing your own chatbot core to power any sort of cognitive response system.

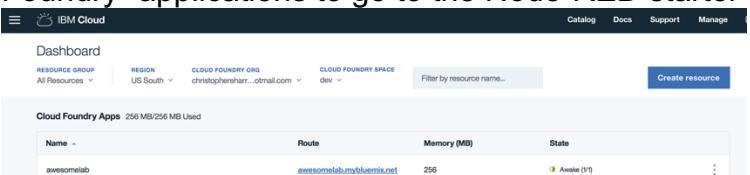
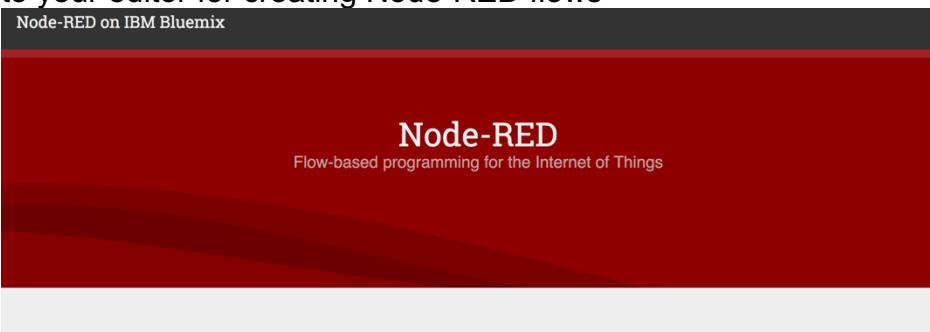
For our purposes, we are going to extend our bot to respond to the spoken word. Lab 4 is about how we use Node-RED to do just that.

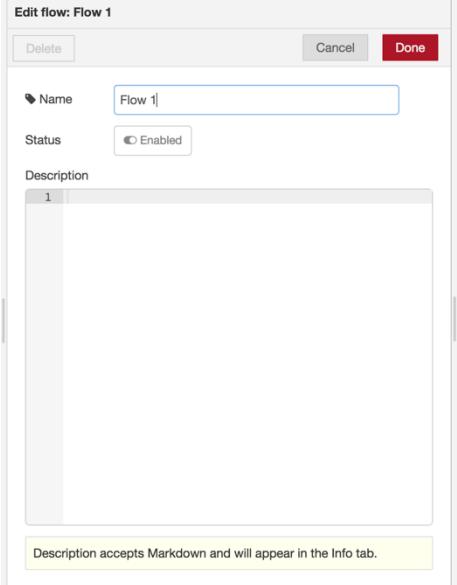


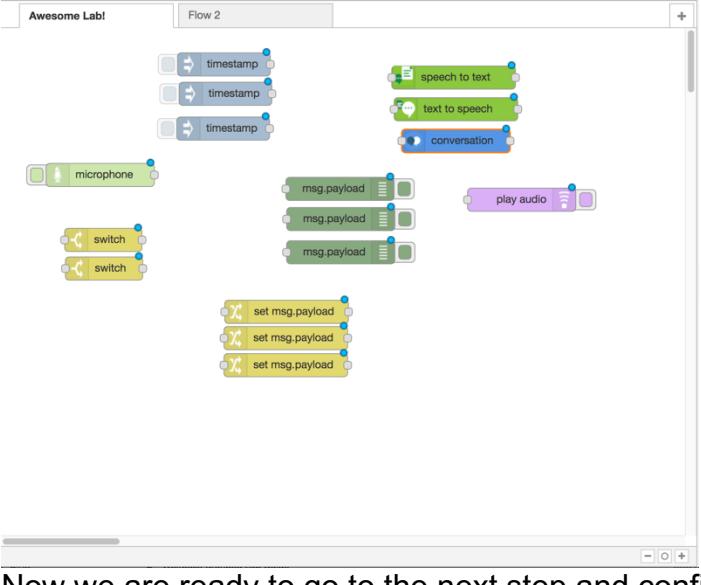
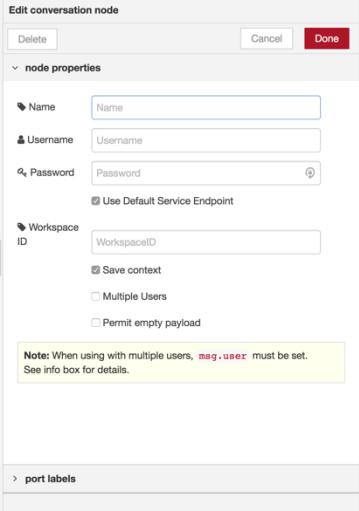
Module 4: Build the Node-RED Flow

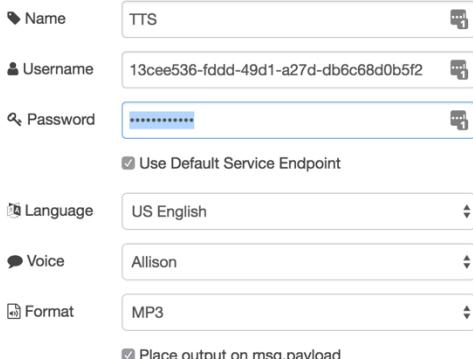
Purpose:	<p>This lab will show you how to construct the Node-RED flow that allows you to interact with our newly created chatbot. When finished with this lab, you will:</p> <ul style="list-style-type: none">• Have a functioning, voice-activated chatbot• Understand the connections and nodes in Node-RED used in the lab• Be able to create a basic Node-RED flow• Be able to configure and wire-up node-RED nodes
Tasks:	<p>Tasks you will complete in this lab exercise include:</p> <ul style="list-style-type: none">• Deploy a Node-RED flow• Interact with Node-RED nodes• Configure and connected Node-RED Nodes• Interact with your chatbot via speech recognition

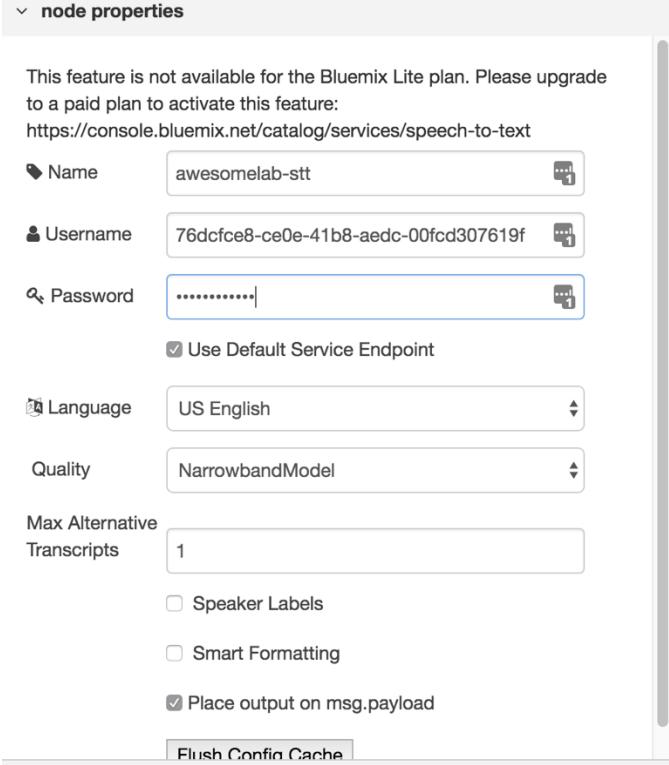
Module 4: Lab Instructions

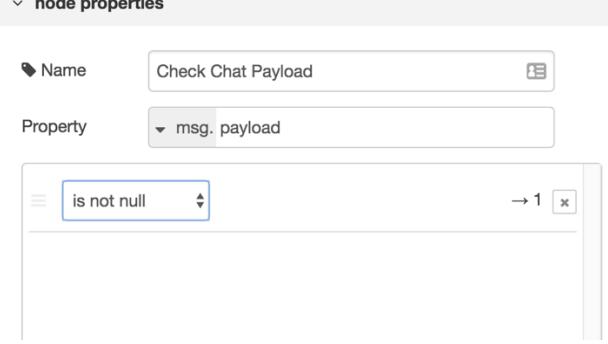
Step	Action
1	<p><u>Deploy the Node-RED nodes needed for this flow</u></p> <p>a. From the main IBM Cloud Dash board, click on the URL in your “Cloud Foundry” applications to go to the Node-RED starter page</p>  <p>b. On the starter page, click on “Go to your Node-RED Flow Editor” to be taken to your editor for creating Node-RED flows</p>  <p>c. You should arrive at the flow editor looking like the image below:</p>  <p>d. First let's rename our flow by double-clicking the “Flow 1” tab and entering a</p>

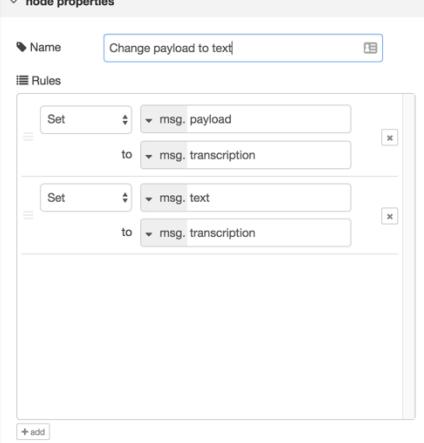
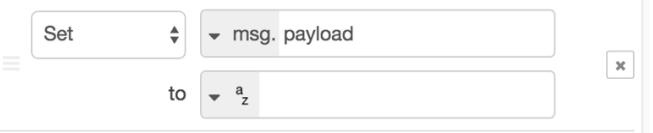
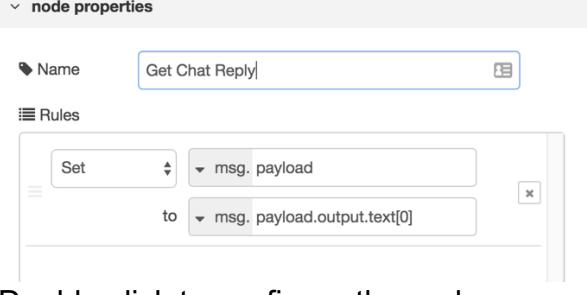
Step	Action
	<p>new name in the resulting dialog box – click “Done” to confirm the change</p>  <p>e. Node-RED flows are made up of “Nodes” which perform actions and then pass the result on to the next Node in line via a mechanism known as “wiring”. Let’s first grab all the Nodes our workflow will use:</p> <ul style="list-style-type: none"> • From the left hand menu, click and drag the following nodes onto the flow editor (we will put them in the correct order in further steps): <ul style="list-style-type: none"> • 3 “Inject” Nodes • 1 “Microphone” Node • 3 “Debug” Nodes • 1 “Play Audio” Node • 2 “Switch” Nodes • 3 “Change” Nodes • 1 “Conversation” Node • 1 “Speech to text” Node • 1 “Text to speech” Node <p>f. The result should be 16 Nodes looking something like the image below:</p>

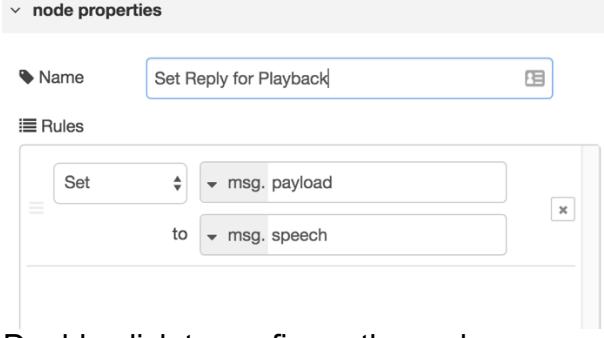
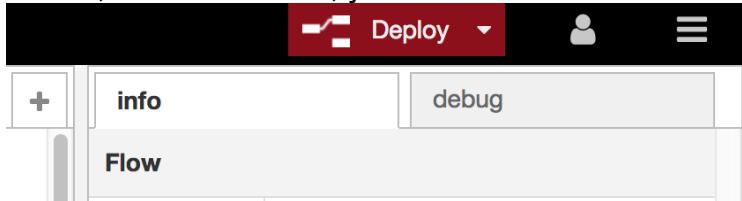
Step	Action
	 <p>g. Now we are ready to go to the next step and configure our Service Nodes</p>
2	<p>Configure the Watson Service Nodes</p> <ol style="list-style-type: none"> <li data-bbox="372 994 1530 1062">a. For service Nodes to communicate with the services, we must configure them so they know how. Use these steps to configure your 3 service nodes: <li data-bbox="372 1062 1057 1100">b. Double click on the blue “conversation” node   <p>c. A dialog pops up from the side with settings options:</p>

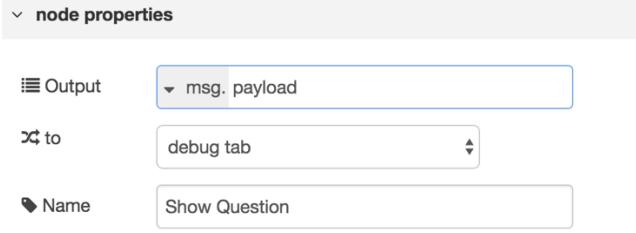
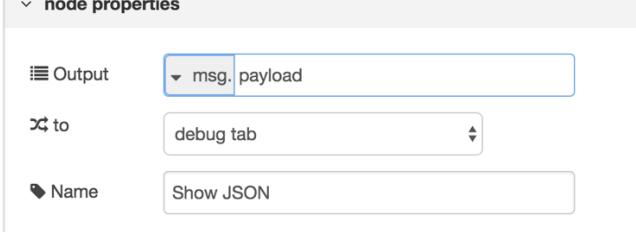
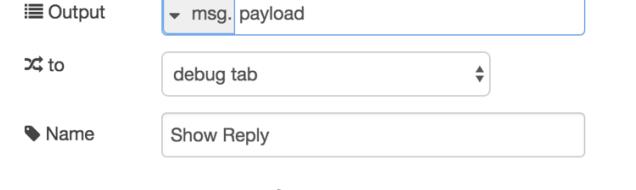
Step	Action
	<ul style="list-style-type: none"> Enter a friendly name – “My Chatbot”, “Awesome Bot” etc... The username and password fields come from the steps we took earlier to get the connection credentials – the username is a very long key and the password is a shorter, random value. (Refer to module 2 – Step 1,G for a review if needed) The workspace ID is what we recorded at the end of Module 3 Once all the information is entered, click “Done” at the top <p>d. Now the Node knows where and how to connect to the conversation service</p> <p>e. Let's do the same for the Text to Speech and Speech to text nodes – double click to modify the node, and follow the settings below:</p> <ul style="list-style-type: none"> Text to Speech:  <ul style="list-style-type: none"> Give it a friendly name Enter the username and password recorded earlier from the service credentials screen Choose the language (English is default) Choose the Voice – this is the voice that will be used to record the responses, and thus is the voice of your bot Choose the format – for the lab use “MP3” Check the “Place output on msg.payload” checkbox Speech to Text:

Step	Action
	 <ul style="list-style-type: none"> • Give it a friendly name • Enter the username and password recorded earlier from the service credentials screen • Change the language (English is default) • Select “NarrowBandModel” from the “Quality options” • Check the “Place output on msg.payload” checkbox • Uncheck “Speaker Labels” and “Speaker Formatting”
3	<p><u>Modify Function Nodes</u></p> <ol style="list-style-type: none"> a. Function nodes are used to modify or validate the messages being passed from node-to node, as well as a host of other functions (like executing external code, creating output files, and a huge volume of custom types that can be added). We are using two types of function nodes: <ul style="list-style-type: none"> • “Switch” Which is a type of compare node • “Change” which is used to reassign values b. Let's configure our two “switch” nodes: <ul style="list-style-type: none"> • Switch Node 1:

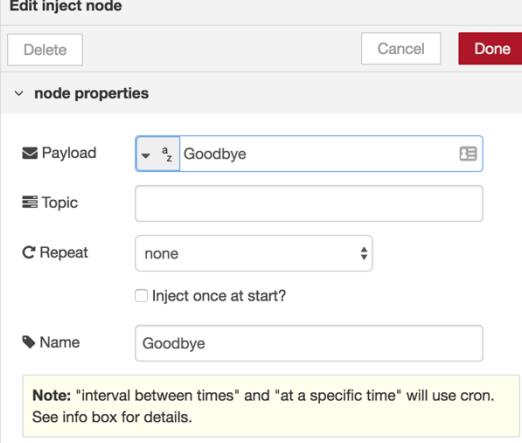
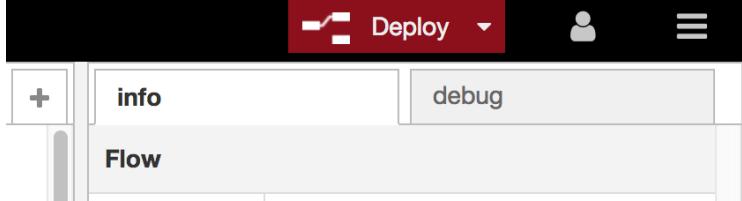
Step	Action
	 <ul style="list-style-type: none"> Double click to configure the node Give it the name “Check Speech to Text” Change the Property from “==” to “is not Null” Click the “+add” button on the lower right to add another comparison Change the property to “is not Null” again Notice that there are a “1” and a “2” next to these comparisons – switches are a way you can split your flow if there are errors as well as comparing and having multiple outputs Click “Done” to commit the changes <p>• Switch Node 2:</p>  <ul style="list-style-type: none"> Double click to configure the node Give it the name “Check Chat Payload” Change the Property from “==” to “is not Null” Click “Done” to commit the changes <p>c. Now let's configure our “Change” nodes – notice that when you drag them to the flow, the text on them is set msg.payload. The msg.payload value is the most common value that is passed from node to node, and can be reset using these functions as needed. Let's configure them to do just that for our flow:</p> <p>• Change Node 1</p>

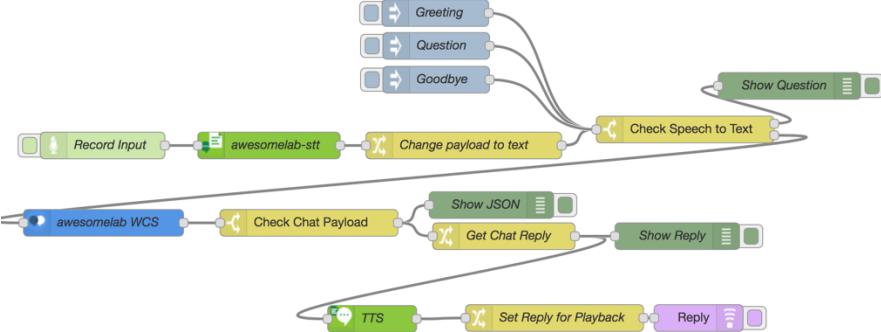
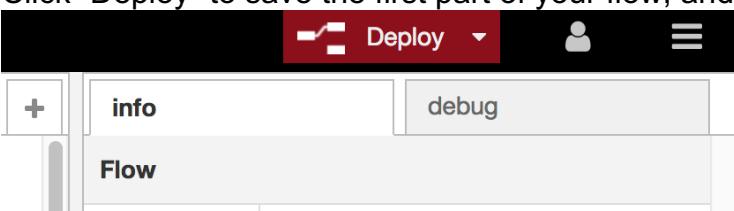
Step	Action
	 <ul style="list-style-type: none"> Double click to configure the node Give it the name “Change Payload to Text” There will be a rule already entered by default – click the “A z” drop down next to the second field, and choose “msg.” from the menu  <ul style="list-style-type: none"> Type “transcription” in the field next to it Click “+add” to add a second rule It will enter msg.payload as a default in the first field – replace “payload” with the word “text” Click the “A z” drop down next to the second field, and choose “msg.” from the menu Type “transcription” in the field next to it (Note – this seems odd to set two fields to the same value, but not doing so results in errors with the Text to Speech service) Click Done to commit the changes <p>Change Node 2</p>  <ul style="list-style-type: none"> Double click to configure the node

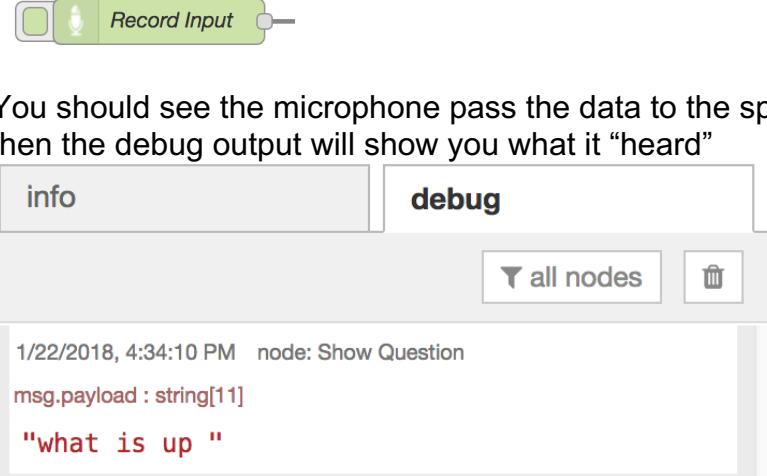
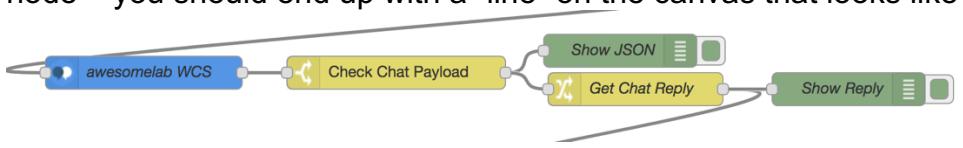
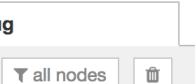
Step	Action
	<ul style="list-style-type: none"> Give it the name “Get Chat Reply” There will be a rule already entered by default – click the “A z” drop down next to the second field, and choose “msg.” from the menu Type “payload.output.text[0]” in the field next to it (Note – the Watson Chatbot returns the results of its “conversation” as a JSON object. The value you are setting above comes from that JSON (which we will expose in another part of the lab)) Click Done to commit the changes <p>• Change Node 3</p>  <ul style="list-style-type: none"> Double click to configure the node Give it the name “Set Reply for Playback” There will be a rule already entered by default – click the “A z” drop down next to the second field, and choose “msg.” from the menu Type “speech” in the field next to it (Note – text to speech returns a file with the text converted to speech – the field we are specifying for msg.payload contains the name of that file, which will get passed to our playback when we wire up the flow) Click Done to commit the changes <p>d. At this point, click “Deploy” in the upper right of the flow editor. Deploy does deploy the code behind this flow into your application, but it also acts somewhat like a “Save” button so that even if your connection breaks, you reboot, a comet strikes, you can resume the flow later.</p>  <p>e. Now let's configure our 3 Debug Nodes – Debug nodes are essentially</p>

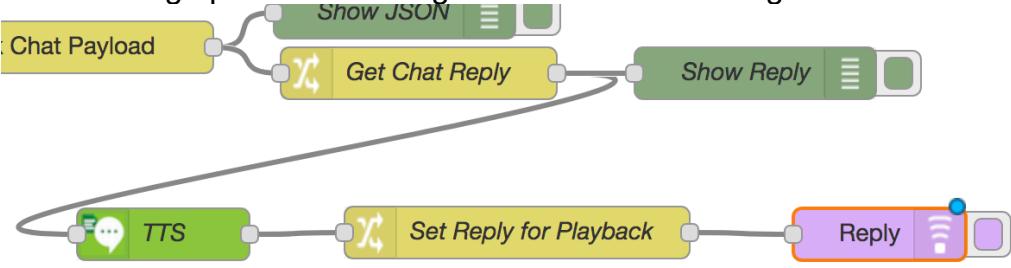
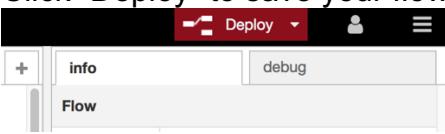
Step	Action
	<p>“output” nodes that show you the information you tell them to (usually msg.payload) so you can “see” what’s being moved around in the flow.</p> <ul style="list-style-type: none"> • Debug Node 1  <ul style="list-style-type: none"> • Double click to configure the node • Give it a name in the “Name” field – “Show Question” for this one • Click “Done” to commit the change • Debug Node 2  <ul style="list-style-type: none"> • Double click to configure the node • Give it a name in the “Name” field – “Show JSON” for this one • Click “Done” to commit the change • Debug Node 3  <ul style="list-style-type: none"> • Double click to configure the node • Give it a name in the “Name” field – “Show Reply” for this one • Click “Done” to commit the change <p>f. Only 3 left to configure! Last let’s configure the “Inject” nodes. Inject nodes simply inject whatever string you tell them into msg.payload – they are great for spot testing complex flows, and in our case, making sure that even if the microphone doesn’t “hear us” very well, we can still test our flow!</p> <ul style="list-style-type: none"> • Inject Node 1

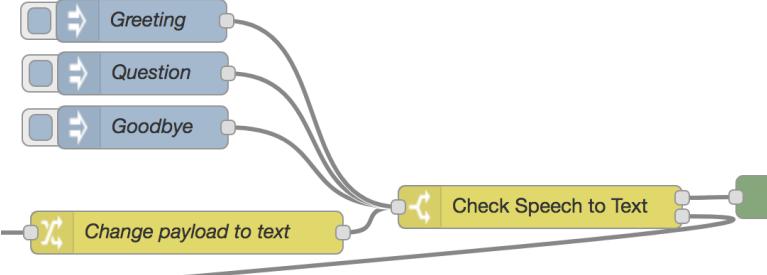
Step	Action
	<p data-bbox="633 280 1188 671"> node properties Payload: Hello, Watson </p> <ul style="list-style-type: none"> Double click to configure the node In the “Payload” field, change the dropdown from timestamp to string Enter the text “Hello Watson” in the “Payload” field Give it a name in the “Name” field – “Greeting” for this one Leave the other fields as their defaults Click “Done” to commit the change <p data-bbox="478 935 731 973">Inject Node 2</p> <p data-bbox="633 973 1188 1453"> Edit inject node Delete Cancel Done </p> <p data-bbox="633 1453 1188 1484"> node properties </p> <p data-bbox="633 1136 1188 1379"> Payload: What time is my flight today? </p> <ul style="list-style-type: none"> Double click to configure the node In the “Payload” field, change the dropdown from timestamp to string Enter the text “What time is the movie” in the “Payload” field Give it a name in the “Name” field – “Question” for this one Leave the other fields as their defaults Click “Done” to commit the change <p data-bbox="478 1717 731 1755">Inject Node 3</p>

Step	Action
	 <ul style="list-style-type: none"> • Double click to configure the node • In the “Payload” field, change the dropdown from timestamp to string • Enter the text “Goodbye” in the “Payload” field • Give it a name in the “Name” field – “Goodbye” for this one • Leave the other fields as their defaults • Click “Done” to commit the change <p>g. Woo Hoo! That's it, you have configured all the nodes. Click “Deploy” up top again for good measure – don't want to lose our progress. Now we can wire the Nodes up and actually have a working flow!</p> 
4	<p><u>Wiring</u></p> <p>a. Wiring is how we tell the Node-RED application what order to do things in. It is as simple as connecting the little gray pips or boxes from one node to the next. When we are done, we should have a flow that resembles the image below:</p>

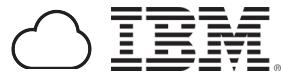
Step	Action
	 <p>b. Let's start from the beginning of our flow, with the "Microphone" input node. Drag it to the top left of the flow canvas (just like reading, flows are best created left-to-right and top-down)</p> <p>c. Drag the "Speech to Text" node next to it, with some space in between</p> <p>d. Drag the "Change Payload to Text" Change node in next to that</p> <p>e. Drag the "Check Speech to Text" Switch node next to that</p> <p>f. Lastly drag the "Show Question" Debug Node next to that</p> <p>g. Now from left to right, simply drag the little gray pip from the node on the left to the one on the right – you will now see a line connecting them, indicating the flow.</p> <p>h. On the "Check Speech to Text" Switch node, you will see two gray pips on the end – use the one on top to connect to the debug node</p>  <p>i. Click "Deploy" to save the first part of your flow, and now you can even test it!</p>  <p>j. On the right hand side, click the "Debug" tab</p> <p>k. Click the little Trash Can icon to remove any messages (if any)</p>  <p>l. Click the small box next to "microphone" and speak into the mic and click it again when done. (You may get a browser security prompt about accessing your mic – simply click "Allow")</p>

Step	Action
	<p>m. You should see the microphone pass the data to the speech to text API, and then the debug output will show you what it “heard”</p>  <p>n. Now let's get that data passed on to the conversation service – Drag the Blue “Watson Conversation” node onto the canvas on a “line” below the current flow (don't worry, the connections can flow down and back)</p> <p>o. Drag the “Check Chat Payload” Change node in next to that – this will have two outputs, so</p> <p>p. Drag the “Show JSON” Debug Node next to that, slightly off-center high so you can</p> <p>q. Drag the “Get Chat Reply” Switch node below the debug node</p> <p>r. Last, drag in the “Show Reply” debug node</p> <p>s. Wiring is as before, from left to right, but start by grabbing the bottom pip on the top line (the “Check Speech to Text” Switch node) and dragging it down and around to the “Watson Conversation” Node.</p> <p>t. Drag from Watson to the Switch node</p> <p>u. Drag from the single pip once to the “Show JSON” Debug node, and then again to the “Get Chat Reply” Change node (you can have multiple actions result from the same node in this way, even if it is not a switch node)</p> <p>v. Lastly, connect the pip from “Get Chat Reply” to the “Show Reply” Debug node – you should end up with a “line” on the canvas that looks like below:</p>  <p>w. Click “Deploy” to save your flow, and now let's test this second part!</p> <p>x. On the right-hand side, click the “Debug” tab (if not already selected)</p> <p>y. Click the little Trash Can icon to remove previous messages (if any)</p> 

Step	Action
	<p>z. Click the microphone, and use a phrase that you trained your chatbot for in the chatbot lab (“Hello, Watson” is always a good default)</p> <p>aa. You should see the flow process the information and print messages out to the “Debug” tab</p> <pre> 1/22/2018, 4:43:17 PM node: Show Question msg.payload : string[13] "hello Watson " 1/22/2018, 4:43:17 PM node: Show JSON msg.payload : Object ▶ { intents: array[1], entities: array[0], input: object, output: object, context: object } 1/22/2018, 4:43:17 PM node: Show Reply msg.payload : string[15] "How you living?"</pre> <p>bb. In the home stretch now! Only a few more nodes to go – grab the “Text to Speech” node and drop it on the canvas below the last line you wired up</p> <p>cc. Grab the “Set Reply for Playback” Change node and drop it next to that on the right</p> <p>dd. Then grab the “Playback” node and drop it in right of that</p> <p>ee. Wire as before, left to right and grabbing the pip from the “Get Chat Reply” change node on the line above and connecting it to the “Text to Speech” Node ending up with something that looks like the image below:</p>  <p>ff. Click “Deploy” to save your flow, and now let’s test this last part!</p>  <p>gg. On the right-hand side, click the “Debug” tab (if not already selected)</p> <p>hh. Click the little Trash Can icon to remove previous messages (if any)</p>

Step	Action
	<p>ii. Click on the microphone and again, use a phrase you trained before – “Hello Watson” is a good testing phrase as always:</p> <pre>1/22/2018, 4:43:17 PM node: Show Question msg.payload : string[13] "hello Watson " 1/22/2018, 4:43:17 PM node: Show JSON msg.payload : Object ▶ { intents: array[1], entities: array[0], input: object, output: object, context: object } 1/22/2018, 4:43:17 PM node: Show Reply msg.payload : string[15] "How you living?"</pre> <p>jj. This time you should see the debug output as before, but you will also hear the voice you selected speak the answer from the conversation bot</p> <p>kk. YAY! It's alive! Now the very last bit of wiring – the Inject Nodes – these nodes are for people (like me) who do not enunciate well – they will ensure that you can pass in a “known good” phrase, just in case your mic isn't working</p> <p>ll. Grab all 3 of the inject nodes and drop them on the canvas just above the “Change Payload to Text” Change node</p> <p>mm. Connect each in turn to the “Check Speech to Text” Switch node</p>  <p>nn. Click “Deploy” to save your flow, and now let's test this very, very last part!</p>  <p>oo. On the right-hand side, click the “Debug” tab (if not already selected)</p>

Step	Action
	<p>pp. Click the little Trash Can icon to remove previous messages (if any)</p>  <p>qq. Click the little blue button on the Inject node(s) to insert the phrase into the flow</p> <p>rr. You should hear the response played, based on what you told your chatbot</p>



Module 4: Lab Summary

Congratulations, your bot is created and trained, and now it converses as well. At this point you have a functioning chat-bot that can interact with voice prompts. While this is a very simplistic bot, all the principals for building a fully functional cognitive assistant are represented here.

From here, only your imagination is the limit as you add new functionality to your assistant and explore the deeper abilities of your bot framework.