

## **HOW TO BUILD A CONVERSATIONAL INTERFACE USING IBM WATSON**

This application (originally developed by IBM's Watson Developer Cloud (<https://github.com/watson-developer-cloud/car-dashboard>)) uses a natural language processing service to analyze text to determine the speaker's intent.

So here's a step-by-step guide for creating a powerful conversational interface with a car using IBM Watson. A machine learning-based tool that can be used for an endless range of uses – from Internet of Things (</internet-of-things/>) devices to building management to mobile apps.

## **SMART SPEECH INTERFACE**

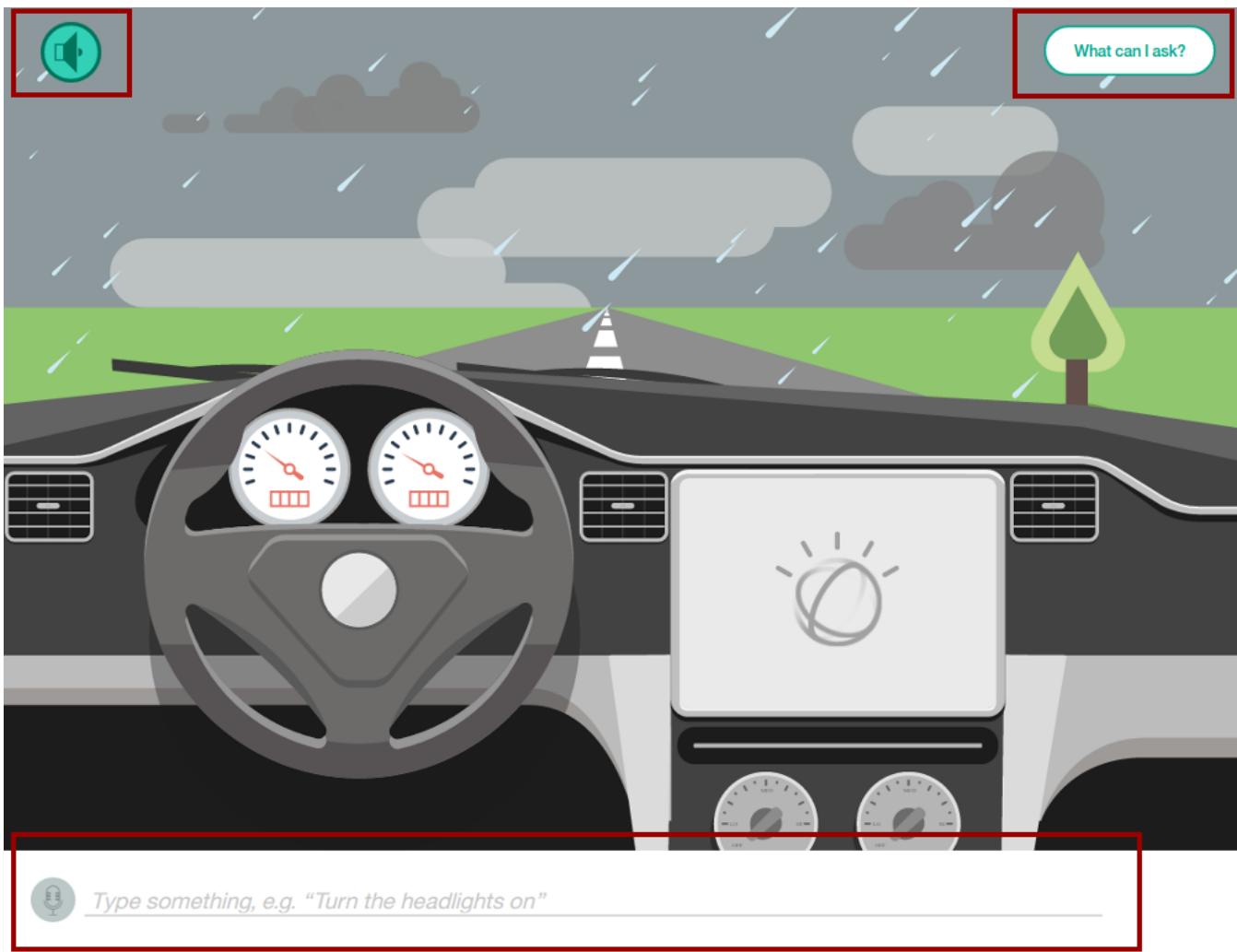
Using a machine learning platform to instantly analyze and determine the intent of a natural language command can provide a wide range of really powerful uses.

- ▶ Step 1: Requirements
- ▶ Step 2: Create the application
- ▶ Step 3: Configure Conversation Service
- ▶ Step 4: Configure Speech-to-Text
- ▶ Step 5: Configure Text-to-Speech
- ▶ Step 6: Fire it up
- ▶ Troubleshooting

**Want to see some source code?** Here's our fork of the application on GitHub (<https://github.com/10xNation/ibm-watson-conversation-service>).

## THE END RESULT

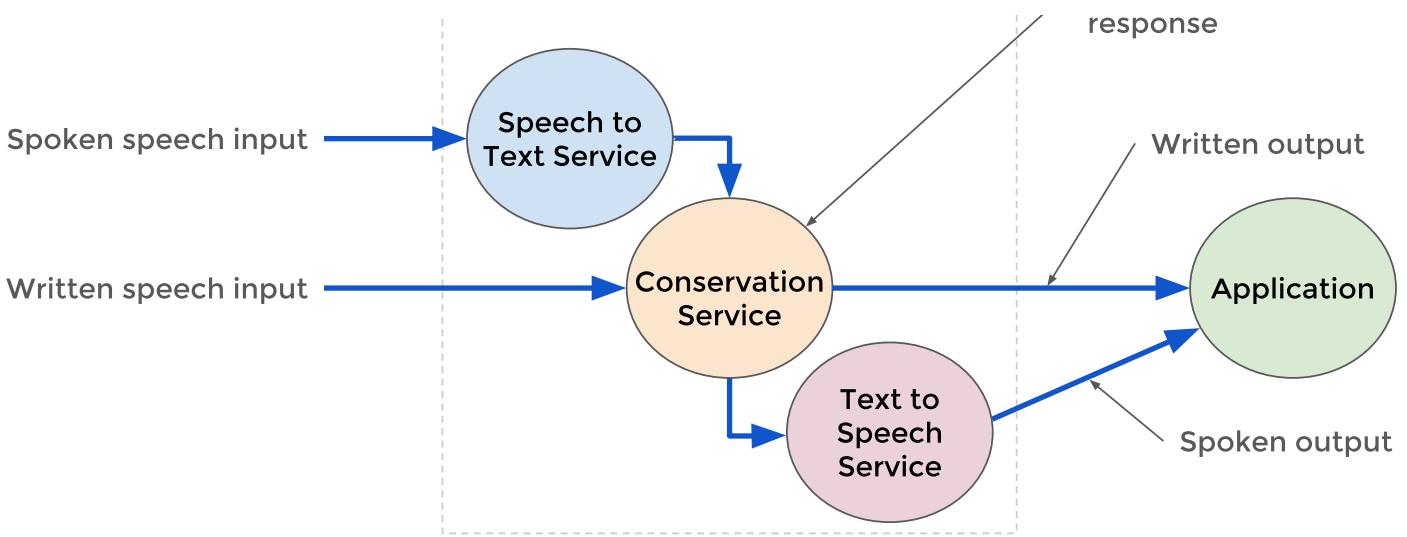
The steps in this guide will create an application similar to the following...



You can also preview a live version (<https://first-conversation-inf528.mybluemix.net/>) of this application. Some of the major highlights include:

- ▶ **Natural language input** – Type or speak in a conversational command, the system figures out your intent
- ▶ **Spoken responses** – Responses are audibly spoken in natural language
- ▶ **Quick input** – Select from a list of intents currently configured in the system

## How it works



This application uses three cloud-based services from IBM Watson:

- ▶ Conversation Service (<http://www.ibm.com/watson/developercloud/conversation.html>) – Natural language processing to automate interactions with end users
- ▶ Speech-to-Text (<https://www.ibm.com/watson/developercloud/speech-to-text.html>) – Convert natural spoken language into text
- ▶ Text-to-Speech (<https://www.ibm.com/watson/developercloud/text-to-speech.html>) – Convert text into audible natural spoken language

**Note:** Most of the following steps can be accomplished through command line or point-and-click. To keep it as visual as possible, this guide focuses on point-and-click whenever possible, but the source code also includes command line scripts if that's your preference.

## STEP 1: REQUIREMENTS

Before we create the Watson services, let's get the system requirements covered.

### Download the source repository.

To start, let's download the source files.

**Note:** You'll need a git client (<https://help.github.com/articles/set-up-git/>) installed on your computer for this step.

Simply move to the directory you want to use for this demo and run the following commands in a terminal...

terminal

```
# Download source repository
git clone https://github.com/10xNation/ibm-watson-conversation-service.git
cd ibm-watson-conversation-service
```

At this point, you can keep the terminal window open and set it aside for now...we'll need it in a later step.

### Name the application.

Right away, let's nail down a name for your new conversational app.

manifest.yml

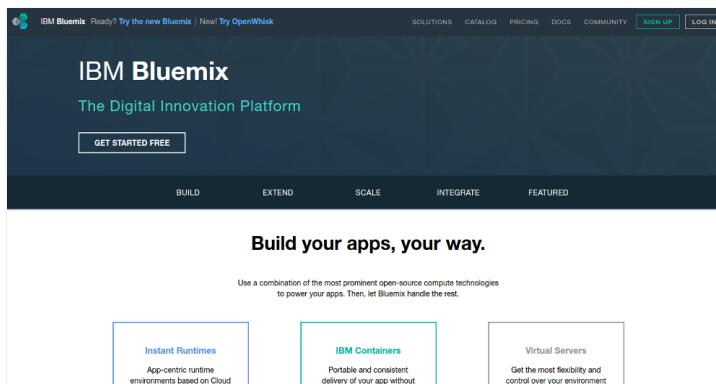
```
...
# Application name
- name:xxxxxxxxxxxxxx
...
```

Replace `xxxxxxxxxxxxxx` in the `manifest.yml` file with a globally unique name for your instance of the application.

The name you choose will be used to create the application's URL — eg. <http://conversation-service-254875.mybluemix.net/>.

### Create a Bluemix account.

Go to the Bluemix Dashboard page (<https://console.ng.bluemix.net/>) (Bluemix is IBM's cloud platform).



If you don't already have one, create a Bluemix account by clicking on the "Sign up" button and completing the registration process.

### Install Cloud-foundry.

A few of the steps in this guide require a command line session, so you'll need to install the Cloud-foundry CLI (<https://github.com/cloudfoundry/cli#downloads>) tool.

### Open a terminal session with Bluemix.

Once the Cloud-foundry CLI tool is installed, you'll be able to log into Bluemix through the terminal.

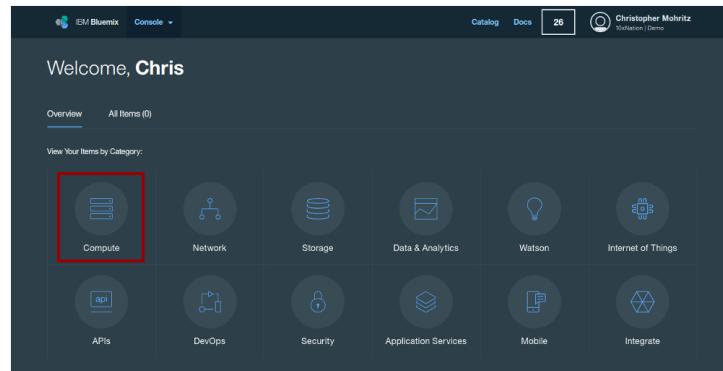
terminal

```
# Log into Bluemix
cf api https://api.ng.bluemix.net
cf login -u YOUR_BLUEMIX_ID -p YOUR_BLUEMIX_PASSWORD
```

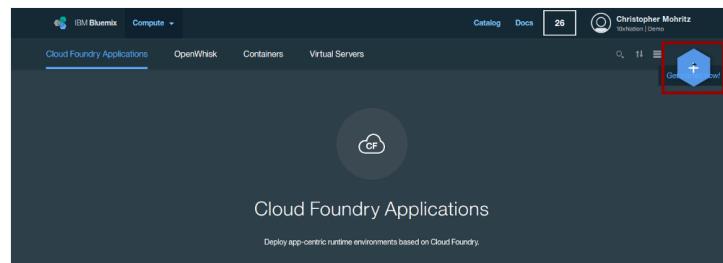
Replace `YOUR_BLUEMIX_ID` and `YOUR_BLUEMIX_PASSWORD` with the respective username and password you created above.

## STEP 2: CREATE THE APPLICATION

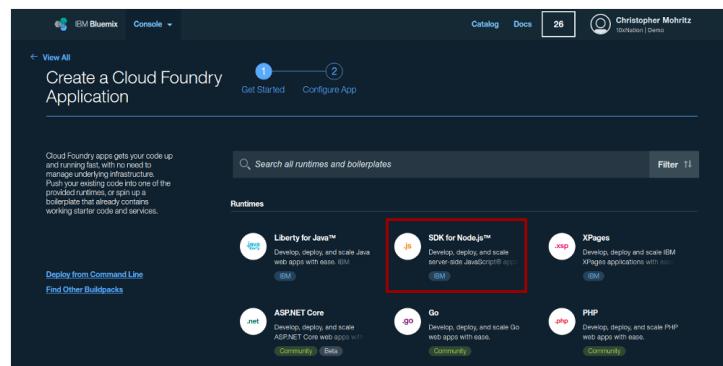
Go to the Bluemix Dashboard page (<https://console.ng.bluemix.net/>).



Once you're signed in and see the Dashboard, click on the "Compute" button.



Then on the next page, click on the "plus" button to add a new application.



In this demo, we'll be using a Node (<https://nodejs.org/>) application, so click on "SDK for Node.js".

SDK for Node.js™

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

View Docs

VERSION 3.x  
TYPE Application  
REGION US South

App name: conversation-service-254875

Host name: conversation-service-254875

Domain: mybluemix.net

Pricing Plans

Plan	Features	Pricing
<input checked="" type="checkbox"/> Default	Run one or more apps free for 30 days (375 GB hours free)	\$0.07 USD/GB-Hour

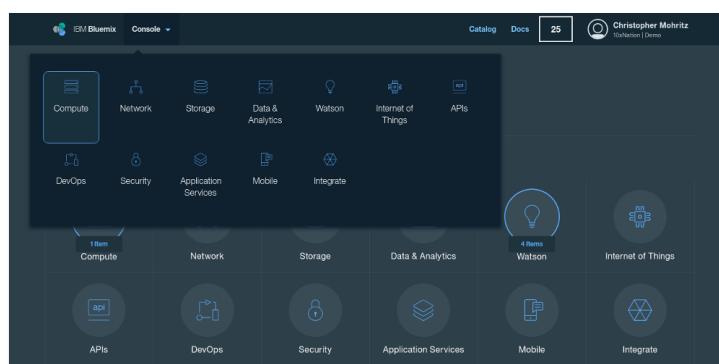
Need Help? Contact Bluemix Sales Estimate Monthly Cost Cost Calculator

Previous Create

Then fill out the information required, using the application name you chose in step #1 – and hit the “Create” button.

### STEP 3: CONFIGURE CONVERSATION SERVICE

To set up your Conversation Service, go to your Bluemix Dashboard page (<https://console.ng.bluemix.net/>).



Hit the navigation dropdown menu and select “Compute.”

Cloud Foundry Applications

conversation-service-254875 Status: Your app is restaging

Getting Started Overview Runtime Connections Logs Monitoring

View App

Conversation-Demo Conversation tree

View Credentials Docs

Connect Existing Connect New

Then click on your application.

The screenshot shows the IBM Bluemix console's Overview tab for a new application. The top navigation bar includes 'Getting Started', 'Overview' (which is selected), 'Runtime', 'Connections', 'Logs', and 'Monitoring'. The main area is divided into several sections: 'Runtime' (Buildpack: Node.js, Instances: 1, Memory Per Instance: 256, Total MB Allocation: 256), 'Connections' (No services connected, 'Connect New' button highlighted with a red box), 'Continuous Delivery' (not configured), and 'Runtime Cost' (\$0). On the right, there's an 'Activity Log' showing recent app starts and updates, and a 'Runtime Cost' summary.

And that should take you to the Overview tab for your application. And since this is a brand new application, you should see a "Create New" button in the Connections widget – click that button.

The screenshot shows the IBM Bluemix Catalog with the 'Watson' category selected. The left sidebar lists categories like All Categories, Storage, Data & Analytics, Watson (selected), Internet of Things, APIs, DevOps, Security, Application Services, Mobile, and Integrate. The main area displays Watson services: AlchemyAPI, Conversation (highlighted with a red box), Document Conversion, Language Translation, Natural Language Classifier, Personality Insights, Retrieve and Rank, Speech to Text, and Text to Speech. Each service has a brief description and an 'IBM' badge.

You should now see a long list of services. Click "Watson" in the Categories filter and then click on "Conversation" to create an instance of that service.

IBM Bluemix    Console ▾

[Catalog](#)   [Docs](#)    Christopher Mohritz  
10xNation | Demo

[View All](#)

## Conversation

Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device. Train Watson Conversation service through an easy-to-use web application, designed so you can quickly build natural conversation flows between your apps and users, and deploy scalable, cost effective solutions.

Service name:  
**Conversation-Demo**

Images

Click an image to enlarge and view screen captures, slides, or videos. Screen caps show the user interface for the service after it has been provisioned.

IBM

Connect to:  
**conversation-service-254875**

[View Docs](#)

AUTHOR IBM  
PUBLISHED 09/16/2016  
TYPE Service  
LOCATION US South

### Pricing Plans

Monthly prices shown are for country or region: [United States](#)

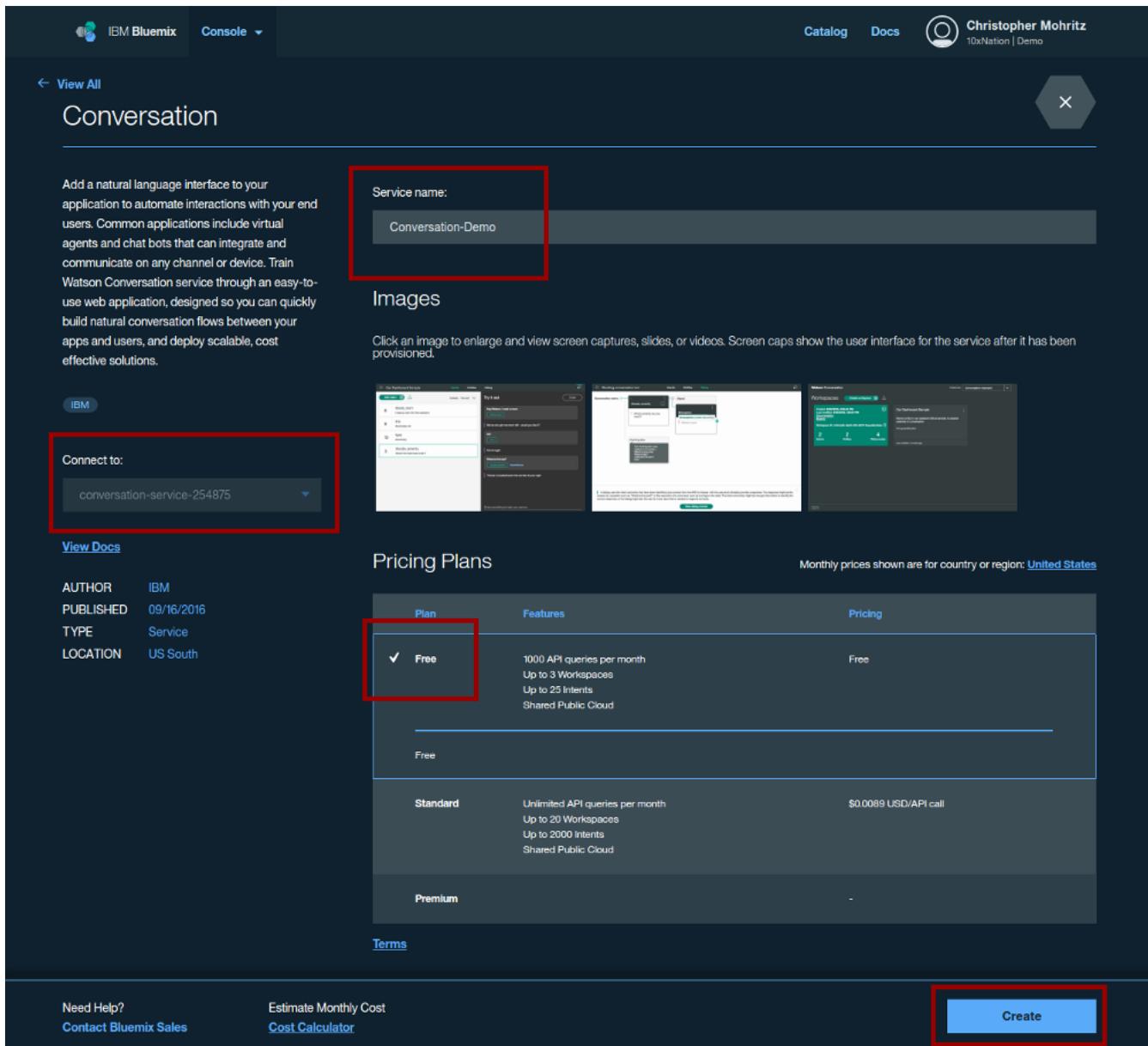
Plan	Features	Pricing
<b>Free</b>	1000 API queries per month Up to 3 Workspaces Up to 25 Intents Shared Public Cloud	Free
Standard	Unlimited API queries per month Up to 20 Workspaces Up to 2000 Intents Shared Public Cloud	\$0.0089 USD/API call
Premium	-	-

[Terms](#)

Need Help?  
[Contact Bluemix Sales](#)

Estimate Monthly Cost  
[Cost Calculator](#)

**Create**



Go ahead and choose a Service Name that makes sense for you — eg. `Conversation-Demo`. For this demo, the “Free” Pricing Plan will do just fine. And by default, you should see your application’s name listed in the “Connected to” field.

**Note:** For whatever reason, once on a while Bluemix won’t give you the option of defining a name for a particular service. If this happens to you, the system will auto-choose a name and it will be shown on your service panel after creating it. This applies to all services you create below.

Click the “Create” button when ready. And enter the Name and Pricing Plan you chose into the `manifest.yml` file.

#### manifest.yml

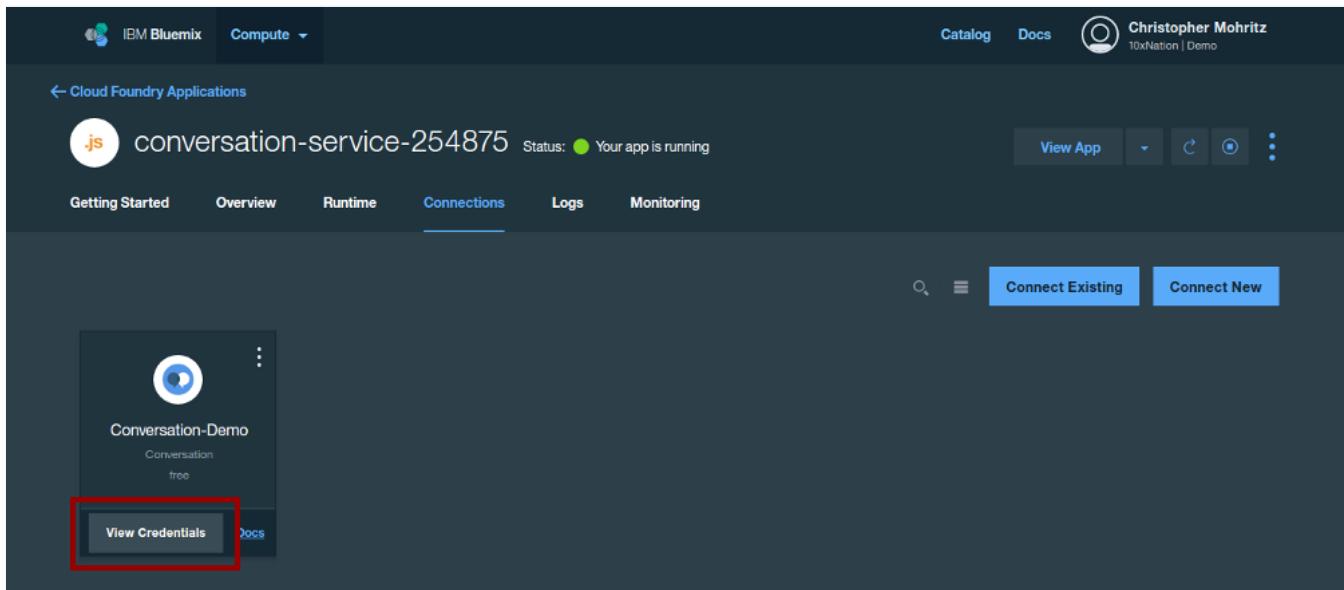
```
...
# Conversation Service
Conversation-Demo:
  label: conversation
  plan: free
...
- services:
  - Conversation-Demo
...
```

If needed, replace both instances of `Conversation-Demo` with your Service Name and `free` with your Pricing Plan.

Feel free to “Restage” your application when prompted.

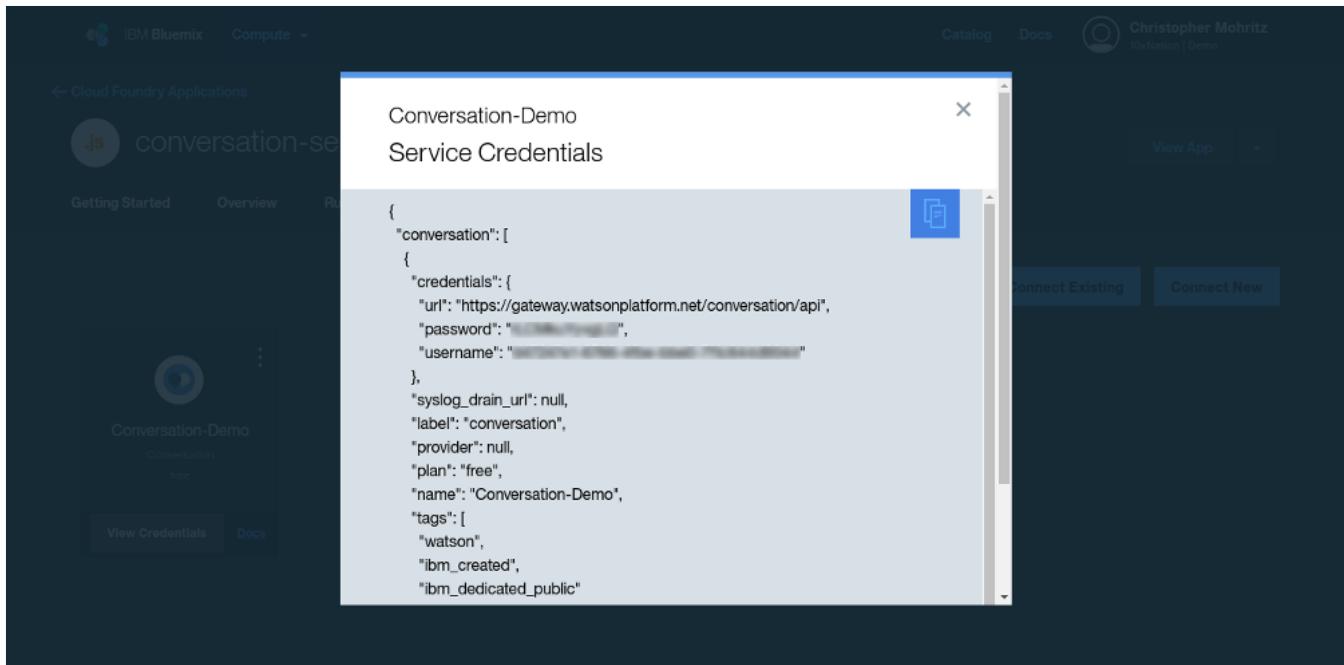
## Enter service credentials.

After your Conversation instance is created, click on the respective “View Credentials” button.



The screenshot shows the IBM Bluemix Cloud Foundry Applications interface. At the top, there are navigation links for 'IBM Bluemix', 'Compute', 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the header, the application 'conversation-service-254875' is listed with a status message 'Status: Your app is running'. The 'Connections' tab is currently selected. On the right side of the card, there are buttons for 'View App', 'Connect Existing', and 'Connect New'. At the bottom left of the card, there is a 'View Credentials' button, which is highlighted with a red box.

And that will pop up a modal with your details.



The screenshot shows a modal window titled 'Conversation-Demo Service Credentials'. The modal contains a JSON object representing the service credentials:

```
{
  "conversation": [
    {
      "credentials": {
        "url": "https://gateway.watsonplatform.net/conversation/api",
        "password": "XXXXXXXXXX",
        "username": "XXXXXXXXXX"
      },
      "syslog_drain_url": null,
      "label": "conversation",
      "provider": null,
      "plan": "free",
      "name": "Conversation-Demo",
      "tags": [
        "watson",
        "ibm_created",
        "ibm_dedicated_public"
      ]
    }
  ]
}
```

Copy/paste your credentials into the respective portion of your `.env` file (in the source code).

```
.env
# Environment variables
...
CONVERSATION_USERNAME=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
CONVERSATION_PASSWORD=xxxxxxxxxxxx
```

Replace `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` with the username listed and `xxxxxxxxxxxx` with the respective password.

## Create a Workspace

The Conversation Service supports multiple Workspaces, so we need to create one for our application.

The screenshot shows the IBM Bluemix Cloud Foundry Applications dashboard. At the top, there are links for 'IBM Bluemix' and 'Compute'. On the right, there are 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the header, a breadcrumb navigation shows 'Cloud Foundry Applications' and the specific application 'conversation-service-254875'. The application name is followed by a status message: 'Status: Your app is running'. To the right of the status are buttons for 'View App', 'Edit', 'Delete', and more options. Below the status, there are tabs for 'Getting Started', 'Overview', 'Runtime', 'Connections' (which is selected), 'Logs', and 'Monitoring'. A red box highlights the 'Connections' section, which displays a card for 'Conversation-Demo'. The card includes a thumbnail, the name 'Conversation-Demo', the type 'Conversation', and the status 'free'. At the bottom of the card are buttons for 'View Credentials' and 'Docs'. To the right of the card are buttons for 'Connect Existing' and 'Connect New'.

Click on your Conversation service.

The screenshot shows the 'Conversation-Demo' service details page. At the top, there are links for 'IBM Bluemix' and 'Watson'. On the right, there are 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the header, a breadcrumb navigation shows 'Conversation-Service-254875' and the specific service 'Conversation-Demo'. The service name is followed by a status message: 'Status: Service available'. To the right of the status are buttons for 'View App', 'Edit', 'Delete', and more options. Below the status, there are tabs for 'Manage' (which is selected), 'Service Credentials', and 'Connections'. A red box highlights the 'Manage' tab. The main content area features a large blue 'Conversation' heading with a purple speech bubble icon. Below the heading, there is a description: 'Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents and chat bots that can integrate and communicate on any channel or device.' To the right of this text is a 'Developer resources:' section with a bulleted list: 'Documentation' and 'Demo'. At the bottom of the page, there is a section titled 'Conversation tooling' with a description: 'Train bots with the Watson Conversation service through an easy-to-use web application. Designed so you can quickly build natural conversation flows between your apps and users, and deploy scalable, cost effective solutions.' A red box highlights the 'Launch tool' button.

On the next page, click on the "Launch tool" button.



Watson Conversation

[Log in with IBM ID](#)[Sign up for IBM ID](#)

Log into the Conversation Tooling site using the same username and password you created for Bluemix in step #1.

Watson Conversation Instances: Conversation-Demo

Create workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

[Create](#) [Import](#)

And let's create a new Workspace. For this demo, we already have a Workspace preconfigured, so click on "Import."

Watson Conversation Instances: Conversation-Demo

Import a workspace

Select a JSON file then choose which elements from the workspace to import.

Choose a file car\_workspace.json

Import

Everything (Intents, Entities, and Dialog)

Intents and Entities

Import

Select the `car_workspace.json` file from the `training` folder in the source code you downloaded in step #1. And import "Everything."

Watson Conversation

Instances: Conversation-Demo

Workspaces [Create](#) [+](#) [↑](#)

Car\_Dashboard

Cognitive Car Dashboard workspace which allows multi-turn conversations to perform tasks in the car.

English (U.S.)

Last modified: just now

After the import, you should see your new Workspace listed on the Workspaces dashboard.

Watson Conversation

Instances: Conversation-Demo

Workspaces [Create](#) [+](#) [↑](#)

Car\_Dashboard

Cognitive Car Dashboard workspace which allows multi-turn conversations to perform tasks in the car.

English (U.S.)

Last modified: just now

[View details](#) [Edit](#) [Duplicate](#) [Download as JSON](#) [Delete](#)

And all we need now is the Workspace ID, and to get that click on the "View Details" link.

Watson Conversation

Instances: Conversation-Demo

Workspaces [Create](#) [+](#) [↑](#)

Created: 9/19/2016, 12:29:57 PM  
Last modified: 9/19/2016, 12:29:57 PM  
[Documentation](#) [Bluemix](#)

Workspace ID: [xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx](#)

14 Intents    8 Entities    61 Dialog nodes

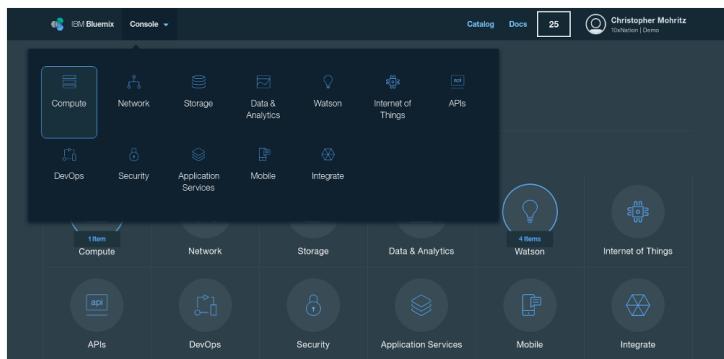
And then copy/paste the Workspace ID into the .env file.:

```
.env
...
# Environment variables
WORKSPACE_ID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
...
```

Your Conversation Service is now ready.

#### STEP 4: CONFIGURE SPEECH-TO-TEXT

Now, let's set up your Speech-to-Text service.



Hit the navigation dropdown menu and select "Compute."

This screenshot shows the IBM Bluemix application details page for 'conversation-service-254875'. The top navigation bar includes links for Catalog, Docs, and a user profile. Below the navigation is a breadcrumb trail: 'Cloud Foundry Applications' &gt; 'conversation-service-254875'. The main content area displays the application name 'conversation-service-254875' with a status of 'Your app is restaging'. Below the name are tabs for 'Getting Started', 'Overview', 'Runtime', 'Connections' (which is underlined in blue), 'Logs', and 'Monitoring'. A red box highlights the application card for 'Conversation-Demo', which includes a thumbnail, the name, a description, and two buttons: 'View Credentials' and 'Docs'. To the right of the application card are two blue buttons: 'Connect Existing' and 'Connect New'. The bottom right corner of the screen shows a set of small navigation icons.

Then click on your application.

This screenshot shows the same IBM Bluemix application details page as the previous one, but with a red box highlighting the 'Connections' tab in the navigation bar. The application name is now 'conversation-service-254875' with a status of 'Your app is running'. The other tabs ('Getting Started', 'Overview', 'Runtime', 'Logs', 'Monitoring') are visible below the 'Connections' tab.

Click on the "Connections" tab – then click on the "Connect New" button.

The screenshot shows the IBM Bluemix Catalog interface. On the left, there's a sidebar with categories like All Categories, Storage, Data & Analytics, Watson (which is highlighted with a blue border), Internet of Things, APIs, DevOps, Security, Application Services, Mobile, and Integrate. The main area displays various Watson services: AlchemyAPI, Conversation, Document Conversion, Language Translation, Natural Language Classifier, Personality Insights, Retrieve and Rank, Speech to Text (which is highlighted with a red border), and Text to Speech. Each service has a brief description and an IBM logo.

Click "Watson" in the Categories filter and then click on "Speech-to-Text" to create an instance of that service.

The screenshot shows the "Speech to Text" service creation page. At the top, it says "Service name:" with "Speech to Text-Demo" entered. Below that is a "Features" section with bullet points for Available Languages, Telephony (narrow-band) models, Keyword Spotting (BETA), and Metadata. A "Connect to:" dropdown is shown with "conversation-service-254875" selected. The "Pricing Plans" section shows a table with three columns: Plan, Features, and Pricing. The "Standard" plan is selected, showing "First thousand minutes are free" and a price of "\$0.02 USD/MINUTE". The "Premium" plan is also listed. At the bottom, there are links for "View Docs", "AUTHOR IBM", "PUBLISHED 09/01/2016", "TYPE Service", "LOCATION US South", and a "Create" button.

Go ahead and choose a Service Name that makes sense for you – eg. Speech to Text-Demo . For this demo, the "Standard" Pricing Plan will do just fine. And by default, you should see your application's name listed in the "Connected to" field.

Click the "Create" button when ready. And enter the Name and Pricing Plan you chose into the `manifest.yml` file.

manifest.yml

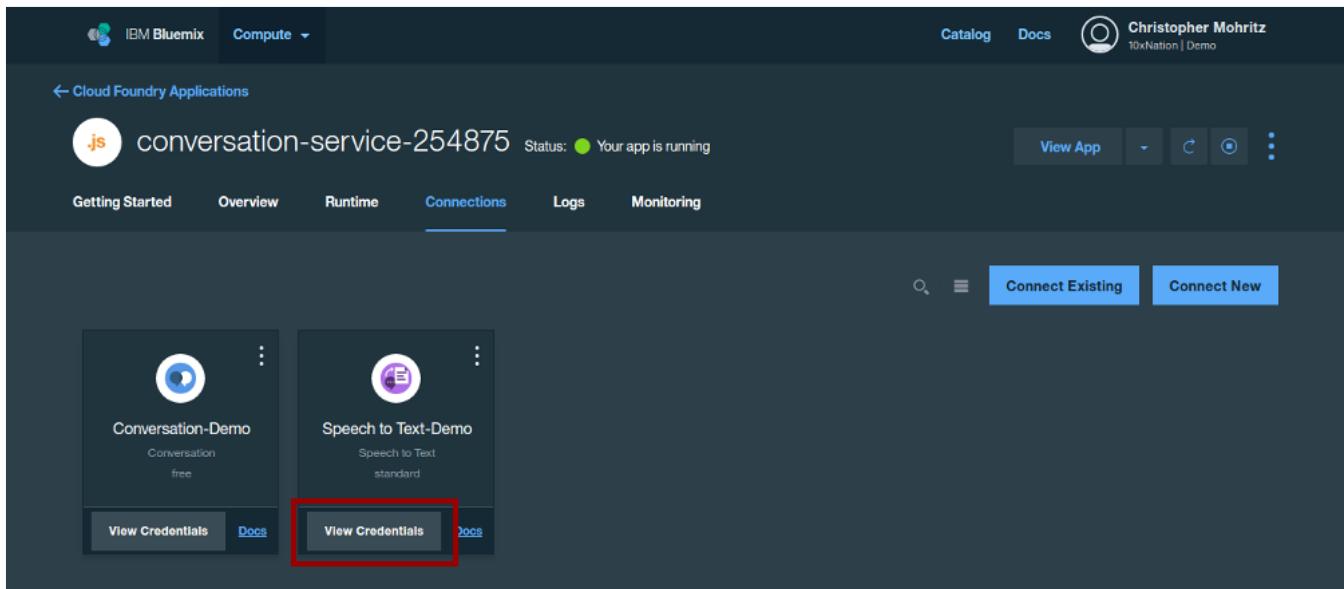
```
...
# Speech to Text
Speech to Text-Demo:
  label: speech_to_text
  plan: standard
...
- services:
...
  - Speech to Text-Demo
...
```

If needed, replace both instances of `Speech to Text-Demo` with your Service Name and `standard` with your Pricing Plan.

Feel free to “Restage” your application when prompted.

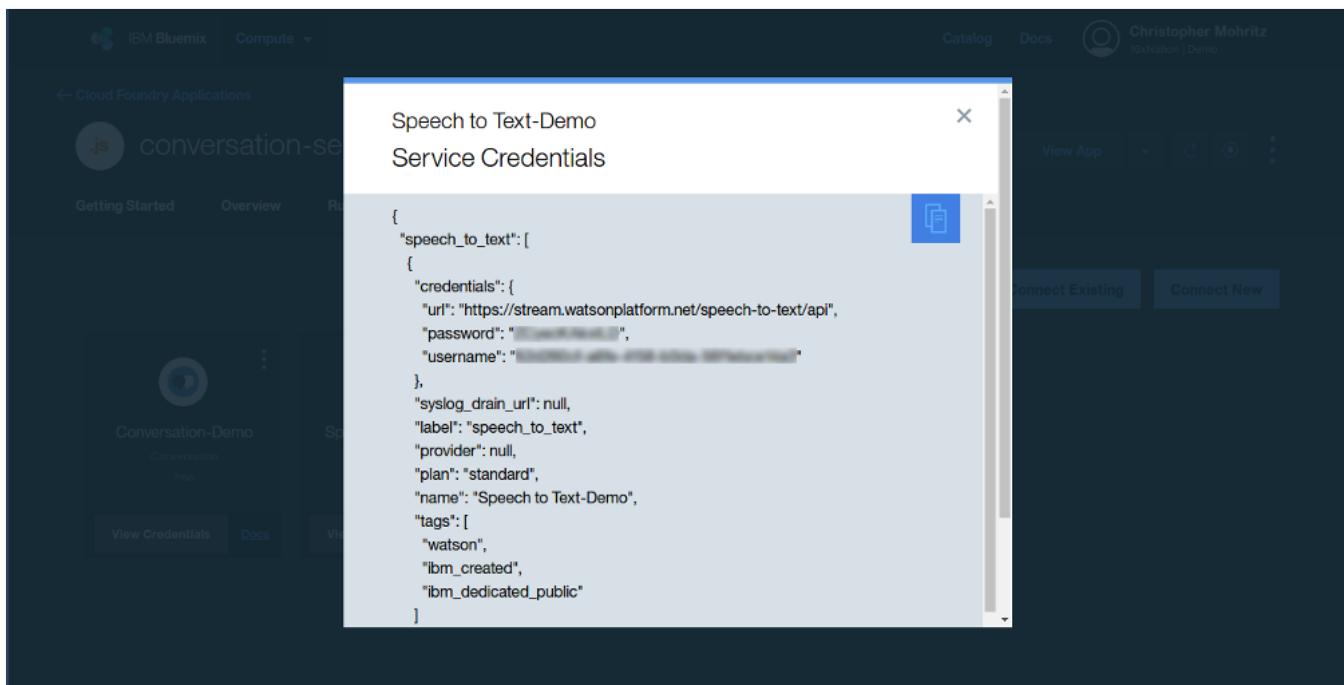
## Enter service credentials.

After your Speech-to-Text instance is created, click on the respective “View Credentials.”



The screenshot shows the IBM Bluemix Cloud Foundry Applications interface. At the top, there's a navigation bar with 'IBM Bluemix', 'Compute', 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the navigation is a header for the application 'conversation-service-254875' with a status indicator 'Your app is running'. The main area has tabs for 'Getting Started', 'Overview', 'Runtime', 'Connections', 'Logs', and 'Monitoring'. Under the 'Connections' tab, there are two service entries. The first is 'Conversation-Demo' with a 'free' plan. The second is 'Speech to Text-Demo' with a 'standard' plan. For each service, there are 'View Credentials', 'Docs', and 'Logs' buttons. The 'View Credentials' button for the 'Speech to Text-Demo' service is highlighted with a red box.

And that will pop up a modal with your details.



The screenshot shows a modal window titled 'Speech to Text-Demo Service Credentials'. Inside the modal, there is a JSON representation of the service credentials:

```
{
  "speech_to_text": [
    {
      "credentials": {
        "url": "https://stream.watsonplatform.net/speech-to-text/api",
        "password": "REDACTED",
        "username": "REDACTED"
      },
      "syslog_drain_url": null,
      "label": "speech_to_text",
      "provider": null,
      "plan": "standard",
      "name": "Speech to Text-Demo",
      "tags": [
        "watson",
        "ibm_created",
        "ibm_dedicated_public"
      ]
    }
  ]
}
```

Copy/paste your credentials into the respective portion of the `.env` file.

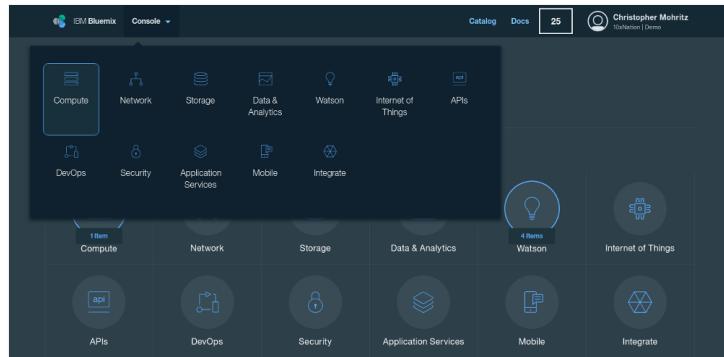
```
.env
...
# Optional: enable Speech to Text
SPEECH_TO_TEXT_USERNAME=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
SPEECH_TO_TEXT_PASSWORD=xxxxxxxxxxxx
...
```

Replace `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` with the username listed and `xxxxxxxxxxxx` with the respective password.

Your Speech-to-Text service is now ready, so let's move onto the next service.

## STEP 5: CONFIGURE TEXT-TO-SPEECH

Now, let's set up your Text-to-Speech service.



Hit the navigation dropdown menu and select "Compute."

Then click on your application.

Click on the "Connections" tab – then click on the "Connect New" button.

The screenshot shows the IBM Bluemix Catalog interface. On the left, there's a sidebar with various service categories: All Categories, Storage, Data & Analytics, Watson (which is highlighted with a blue border), Internet of Things, APIs, DevOps, Security, Application Services, Mobile, and Integrate. The main area displays several Watson services with their icons and brief descriptions. One service, "Text to Speech", is highlighted with a red box.

Service	Description
AlchemyAPI	An AlchemyAPI service that analyzes your unstructured text and image
Conversation	Add a natural language interface to your application to automate
Document Conversion	Converts a HTML, PDF, or Microsoft Word™ document into a normalized
Language Translation	Translate text from one language to another for specific domains.
Natural Language Classifier	Natural Language Classifier performs natural language classification on
Personality Insights	The Watson Personality Insights derives insights from transactional
Retrieve and Rank	Add machine learning enhanced search capabilities to your application
Speech to Text	Low-latency, streaming transcription
<b>Text to Speech</b>	Synthesizes natural-sounding speech from text.

Click “Watson” in the Categories filter and then click on “Text-to-Speech” to create an instance of that service.

This screenshot shows the details page for the "Text to Speech" service in the IBM Bluemix catalog. At the top, there's a "Service name:" input field containing "Text to Speech-Demo", which is also highlighted with a red box. Below it, there's a "Connect to:" dropdown menu with "conversation-service-254875" selected, also highlighted with a red box. The page includes sections for "Features" (listing various languages and voices) and "Pricing Plans". The "Standard" plan is selected and highlighted with a red box. At the bottom, there are links for "Need Help?", "Estimate Monthly Cost", and "Create" (which is also highlighted with a red box).

Plan	Features	Pricing
✓ Standard	The first million characters are free You will be charged per thousand characters	\$0.02 USD/THOUSAND CHAR
Premium	-	-

Go ahead and choose a Service Name that makes sense for you – eg. `Text to Speech-Demo`. For this demo, the “Standard” Pricing Plan will do just fine. And by default, you should see your application’s name listed in the “Connected to” field.

Click the “Create” button when ready. And enter the Name and Pricing Plan you chose into the `manifest.yml` file.

#### `manifest.yml`

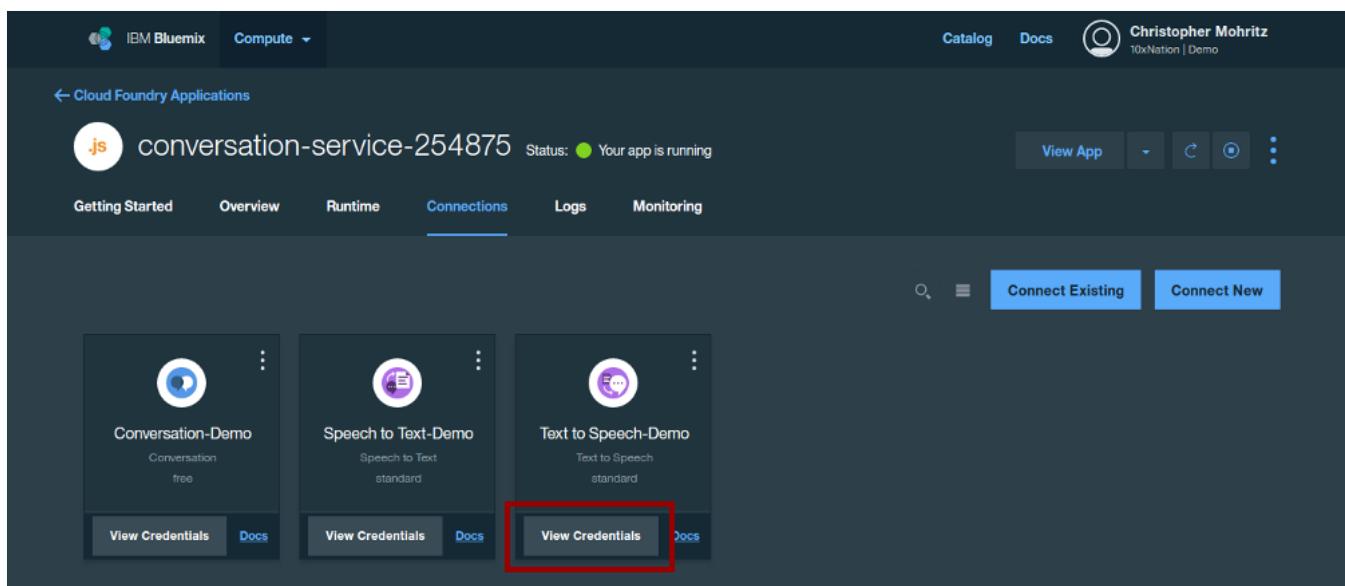
```
...
# Text to Speech
Text to Speech-Demo:
  label: text_to_speech
  plan: standard
...
- services:
...
- Text to Speech-Demo
...
```

If needed, replace both instances of `Text to Speech-Demo` with your Service Name and `standard` with your Pricing Plan.

Feel free to “Restage” your application when prompted.

#### Enter service credentials.

After your Text-to-Speech instance is created, click on the respective “View Credentials” button.



The screenshot shows the IBM Bluemix Compute interface. At the top, there's a navigation bar with 'IBM Bluemix' and 'Compute' dropdowns, 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the navigation is a breadcrumb 'Cloud Foundry Applications' and the application name 'conversation-service-254875'. The status is shown as 'Your app is running'. The main area has tabs for 'Getting Started', 'Overview', 'Runtime', 'Connections' (which is selected), 'Logs', and 'Monitoring'. Under the 'Connections' tab, three service instances are listed in cards:

- Conversation-Demo**: Conversation free. Buttons: View Credentials (highlighted with a red box), Docs.
- Speech to Text-Demo**: Speech to Text standard. Buttons: View Credentials, Docs.
- Text to Speech-Demo**: Text to Speech standard. Buttons: View Credentials (highlighted with a red box), Docs.

Below the cards are buttons for 'Connect Existing' and 'Connect New'.

And that will pop up a modal with your details.

The screenshot shows the IBM Bluemix Cloud Foundry Applications interface. A modal window titled "Text to Speech-Demo Service Credentials" displays the following JSON code:

```
{
  "text_to_speech": [
    {
      "credentials": {
        "url": "https://stream.watsonplatform.net/text-to-speech/api",
        "password": "xxxxxxxxxxxx",
        "username": "xxxxxxxxxxxx"
      },
      "syslog_drain_url": null,
      "label": "text_to_speech",
      "provider": null,
      "plan": "standard",
      "name": "Text to Speech-Demo",
      "tags": [
        "watson",
        "ibm_created",
        "ibm_dedicated_public"
      ]
    }
  ]
}
```

Copy/paste your credentials into the respective portion of the `.env` file.

```
.env
...
# Optional: enable Speech to Text
TEXT_TO_SPEECH_USERNAME=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
TEXT_TO_SPEECH_PASSWORD=xxxxxxxxxxxx
...
...
```

**PRE-LAUNCH PREVIEW**  
PLEASE PARDON OUR DUST  
*we're still building*

Replace `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` with the username listed and `xxxxxxxxxxxx` with the respective password.

Your Text-to-Speech service is now ready, so let's move onto the next service.

## STEP 6: FIRE IT UP

Nearly everything is ready to go, so let's put the application to work.

### Set the application memory.

First, let's give the application a little more memory to work with.

The screenshot shows the IBM Bluemix console navigation bar. The "Compute" icon is highlighted in the dropdown menu, indicating it is selected.

Hit the navigation dropdown menu and select "Compute."

The screenshot shows the IBM Bluemix Cloud Foundry Applications interface. At the top, there are navigation links for 'IBM Bluemix', 'Compute', 'Catalog', 'Docs', and a user profile for 'Christopher Mohritz'. Below the header, the application 'conversation-service-254875' is displayed with a status message: 'Status: C Your app is restaging'. The application icon is a white circle with a blue 'js' logo. The application name is 'conversation-service-254875'. The status is 'Your app is restaging'. Below the application details, there are tabs for 'Getting Started', 'Overview', 'Runtime', 'Connections', 'Logs', and 'Monitoring'. On the right side, there are buttons for 'View App', 'Connect Existing', and 'Connect New'. A red box highlights the application icon and name.

Then click on your application.

The screenshot shows the IBM Bluemix Cloud Foundry Applications interface for the 'conversation-service-254875' application. The 'Runtime' tab is selected. The application is running, as indicated by the green status bar. The runtime settings include: BUILDPACK (SDK for Node.js™), INSTANCES (1), MBS PER INSTANCE (512), and Total MB Allocation (512). A red box highlights the 'Save' button in the MBS PER INSTANCE section.

Click on the “plus” sign for “MBs per Instance” – set it to 512 MB – then hit “Save.”

## Launch the application.

To bring the application to life, simply run the following command – making sure the terminal is in the repository directory and logged into Bluemix.

```
terminal  
cf push
```

This command will upload the files, configure your new application and start it.

**Note:** You can use the same `cf push` command to update the same application after it's originally published.

## Take a look.

After the server has started, you'll be able to open the application in your browser at the respective URL.

IBM Bluemix Compute

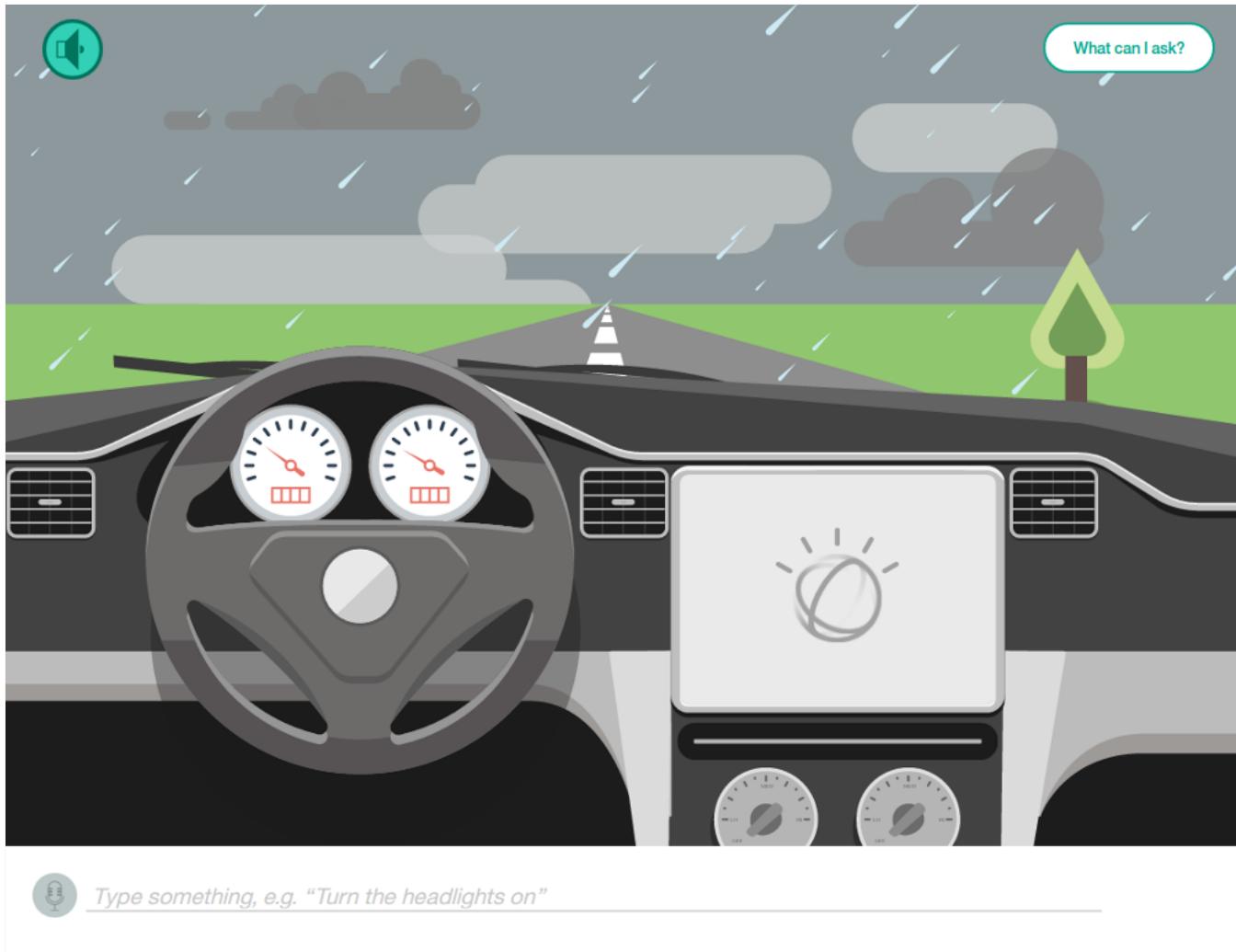
Catalog Docs Christopher Mohritz  
10xNation | Demo

Cloud Foundry Applications OpenWhisk Containers Virtual Servers

512 MB / 512 GB Used

conversation-service-2...  
conversation-service-254875.m...  
Running C ⚡ Open URL

The application should look something like this...



## Add intents

Open the "intents" tab in your "workspace" and follow the tutorial on this page:  
<https://www.ibm.com/watson/developercloud/doc/conversation/getting-started.html>  
to create new intents.