Nishava, Inc. Deep Azure@McKesson

Handed out: 01/02/2018 Due by 11:00 PM CST on Saturday, 2/10/2018

**Final Project, Assignment 15**

Table of Contents

[Problem Statement 2](#_Toc506074738)

[Description of Hardware 2](#_Toc506074739)

[Description of Software 2](#_Toc506074740)

[Data and URL of Data set used 3](#_Toc506074741)

[Implementation 3](#_Toc506074742)

[ Prepare data 3](#_Toc506074743)

[ Upload data 4](#_Toc506074744)

[ Create Resource Group 10](#_Toc506074745)

[ Create Azure Web Bot 11](#_Toc506074746)

[ Associate Web Bot with QnA Maker 13](#_Toc506074747)

[ Add a Disclaimer for the customers to accept 14](#_Toc506074748)

[ Connecting to Facebook Messenger 22](#_Toc506074749)

[ Connecting to Skype 30](#_Toc506074750)

[2. From Skype, search for KaviPharmaBot and send a few messages to test 32](#_Toc506074751)

[ Connecting to Bot through a Webpage 33](#_Toc506074752)

[Summary 35](#_Toc506074753)

[Lesson Learned 35](#_Toc506074754)

[Advantages of Azure Bot Services 35](#_Toc506074755)

[Disadvantages of Azure Bot Services 35](#_Toc506074756)

[Next Steps 35](#_Toc506074757)

[References 36](#_Toc506074758)

[YouTube URLs, GitHub URL 36](#_Toc506074759)

[ Short video: 36](#_Toc506074760)

[ Long video: 36](#_Toc506074761)

[ GitHub Repository 36](#_Toc506074762)

# Problem Statement

Nowadays in many countries, Hospitals and Pharmacies are busy and sometimes just to have a few questions answered can be very time consuming. This situation causes people to postpone looking into their health issues until they become serious which is unfortunately sometimes too late. While increasing the staff could be an easy answer to this problem, it is not always feasible due to the cost associated with more staff members. One of the solutions could be the use of Azure Web Bot across different platforms (WebPage, Facebook Messenger, Skype) to answer most of the quick and easy questions that customers have, advise the customers to go to a pharmacy or hospital for further investigation or to go to the nearest hospital urgently. Another advantage is that the Customers knows that the information provided by the bots are updated recommendations of local Health Professionals as opposed to a general article on the internet.

# Description of Hardware

Computer with Core i7 @2.60GHz and 16GB RAM

IPhone 7

# Description of Software

Windows 10 64 bit

Facebook Messenger

Skype

Google Chrome

Powershell

Microsoft Word 2016

Microsoft Account with Free trial for Azure

# Data and URL of Data set used

In this scenario, we will focus on one health concern, which causes multiple busy periods at hospitals and pharmacies during the year: the common cold.

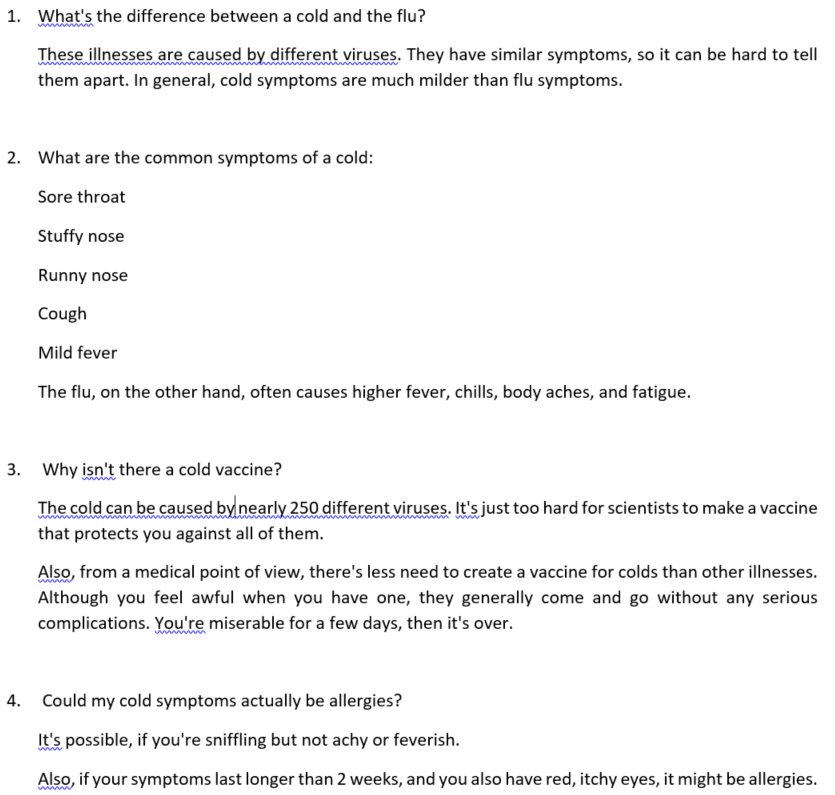
For the purpose of this project, the data from the link below will be used to represent the recommendations of the local Health Professionals.

<https://www.webmd.com/cold-and-flu/top-10-questions-cold>

# Implementation

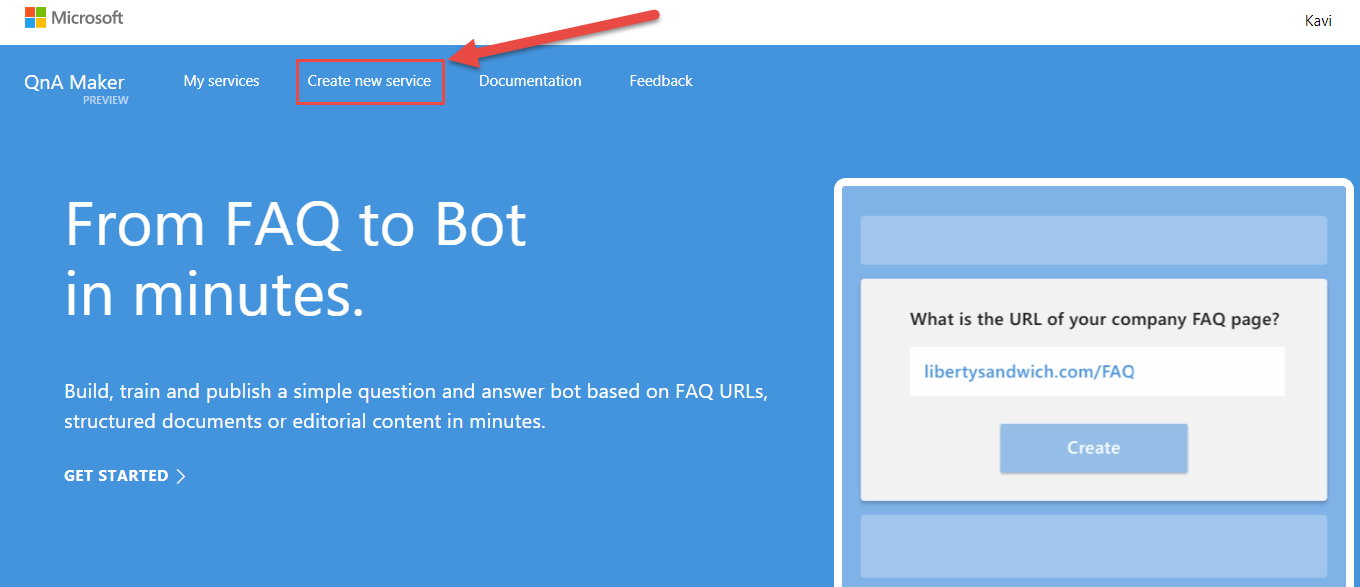
## Prepare data

1. The questions and answers were captured in a Microsoft Word document and broken down to form shorter answers to better simulate a human conversation.
2. Extract of the document:

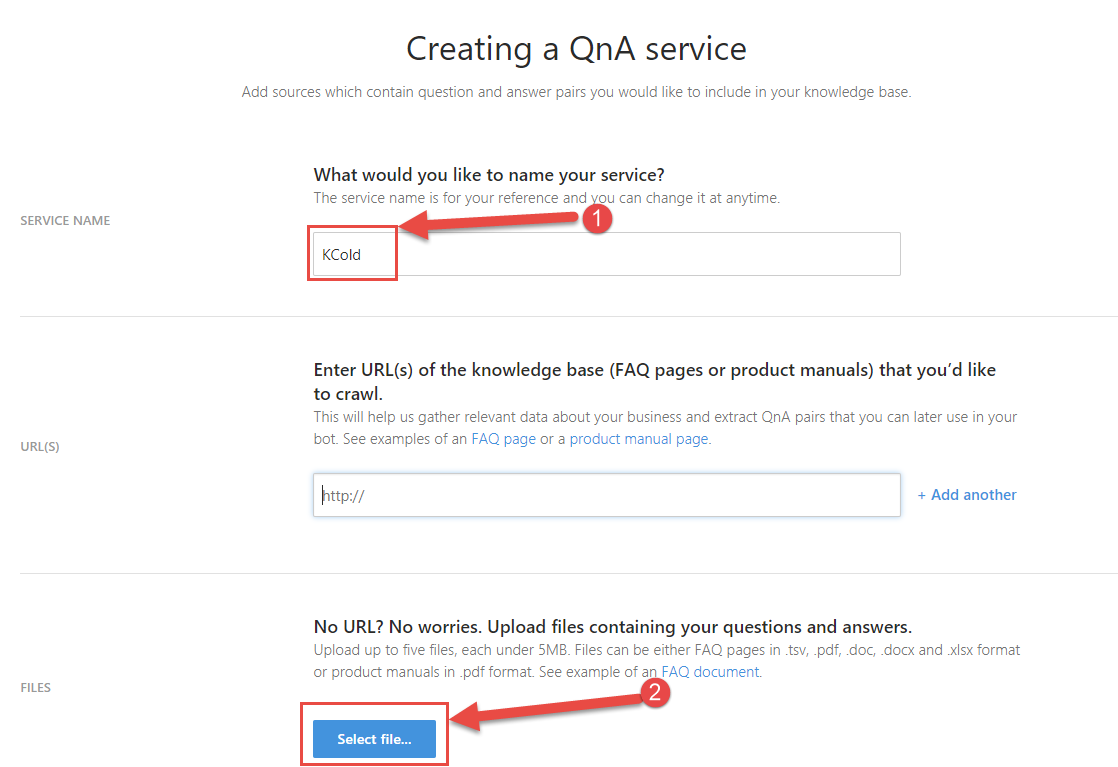


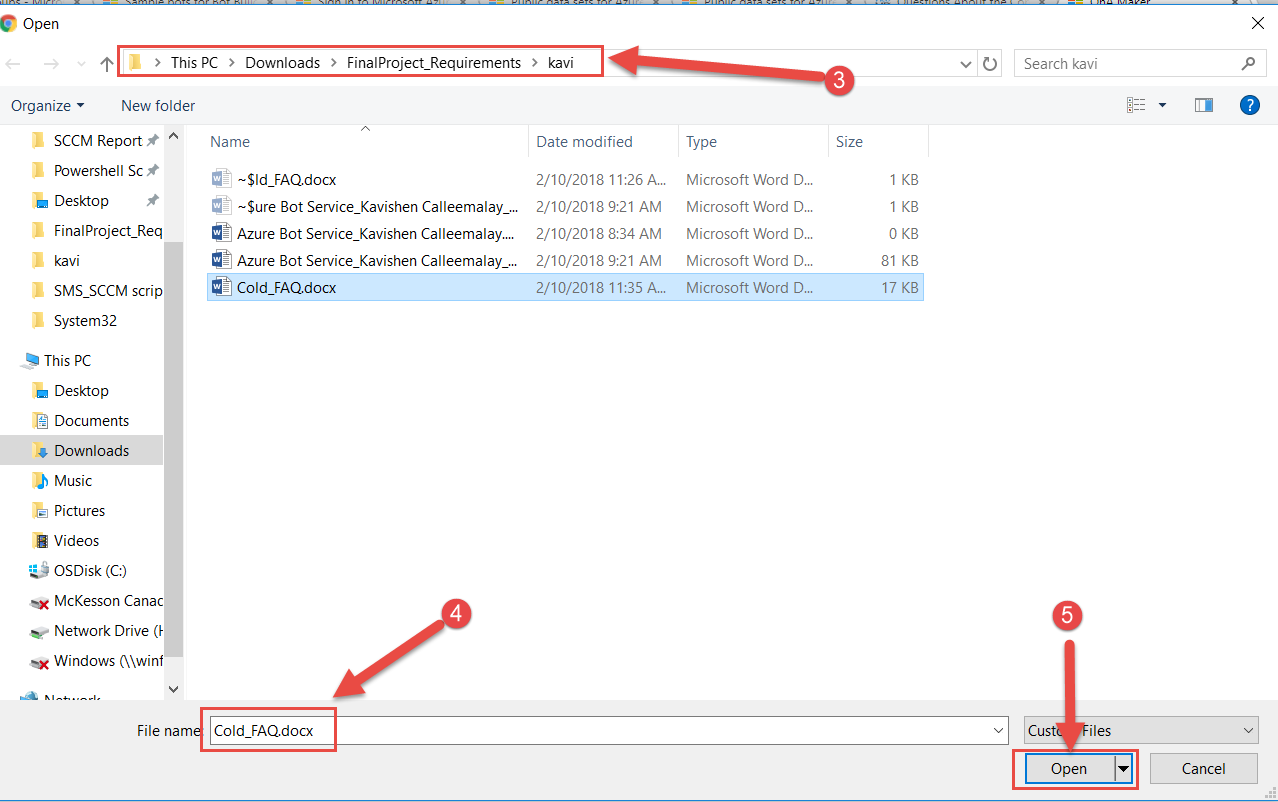
## Upload data

1. Login on QNA Maker (<https://qnamaker.ai>) using your Microsoft account
2. Click on **Create new service**

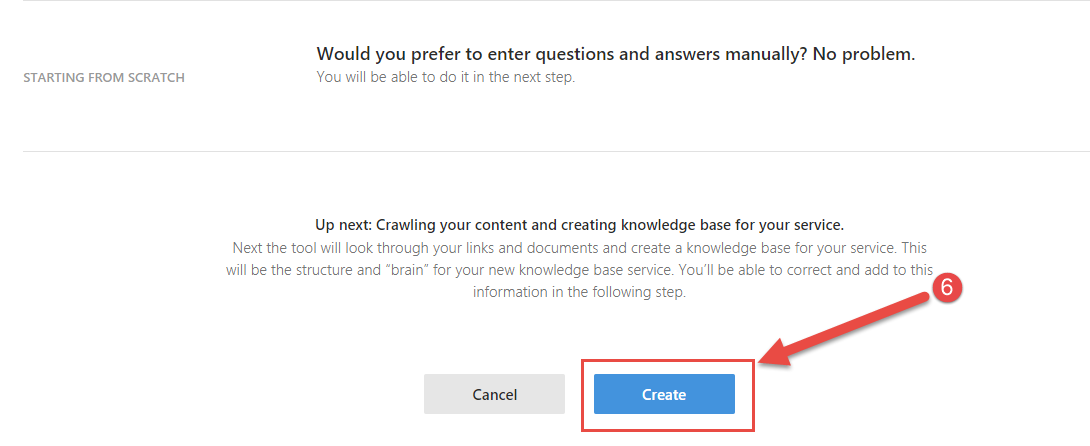


1. Type in a **Service Name** and browse to the FAQ word document created above

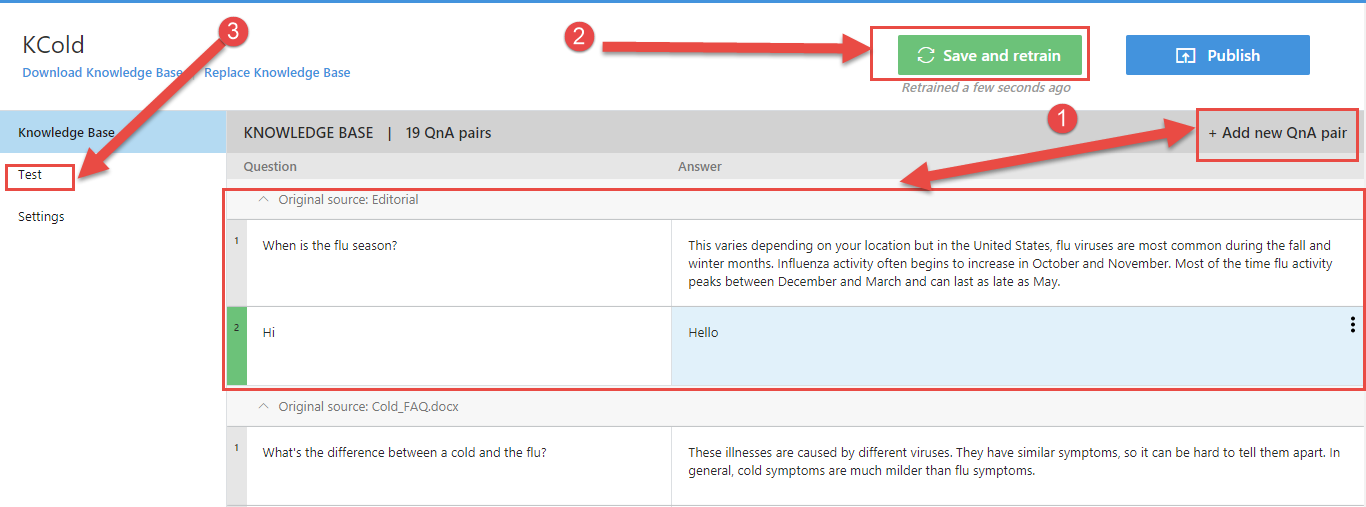




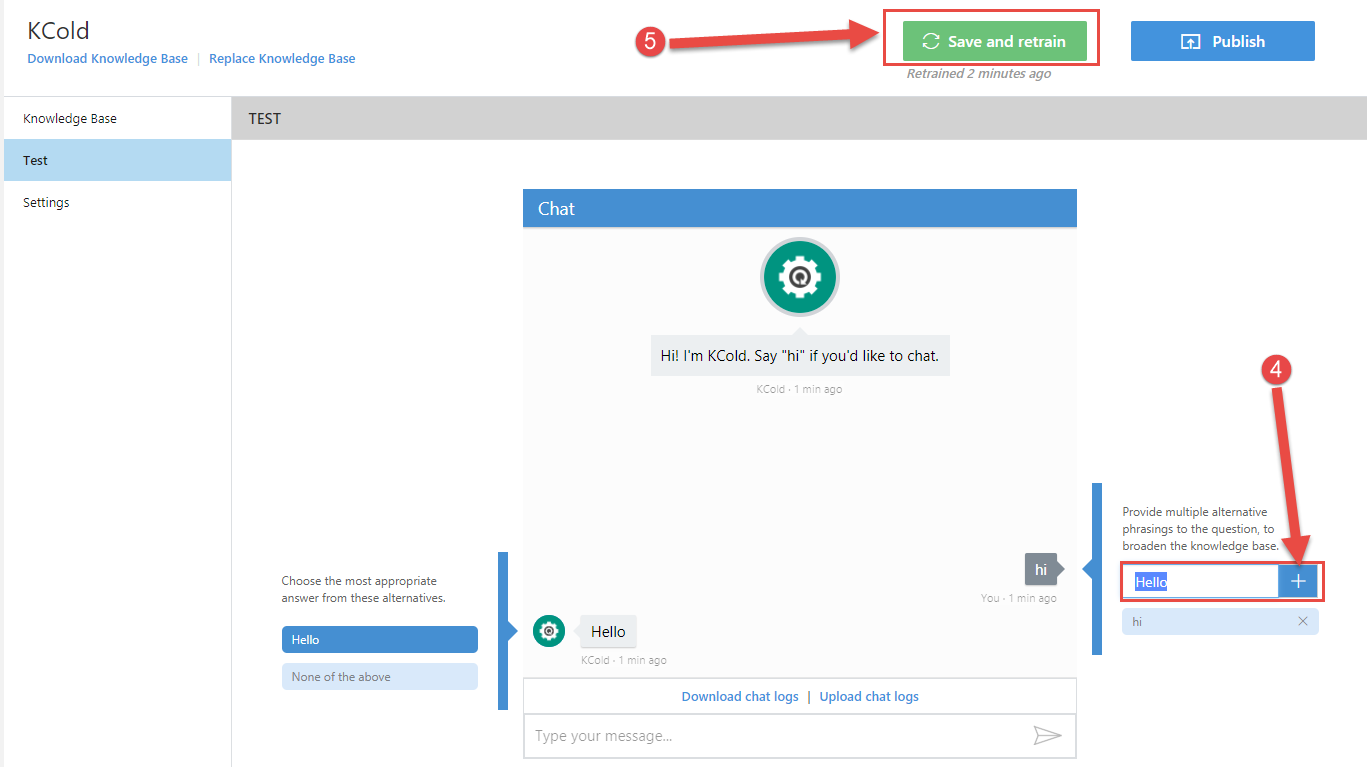
1. Scroll down and click on **Create**



1. Add new entries, update the Knowledge Base entries and test the responses to the questions



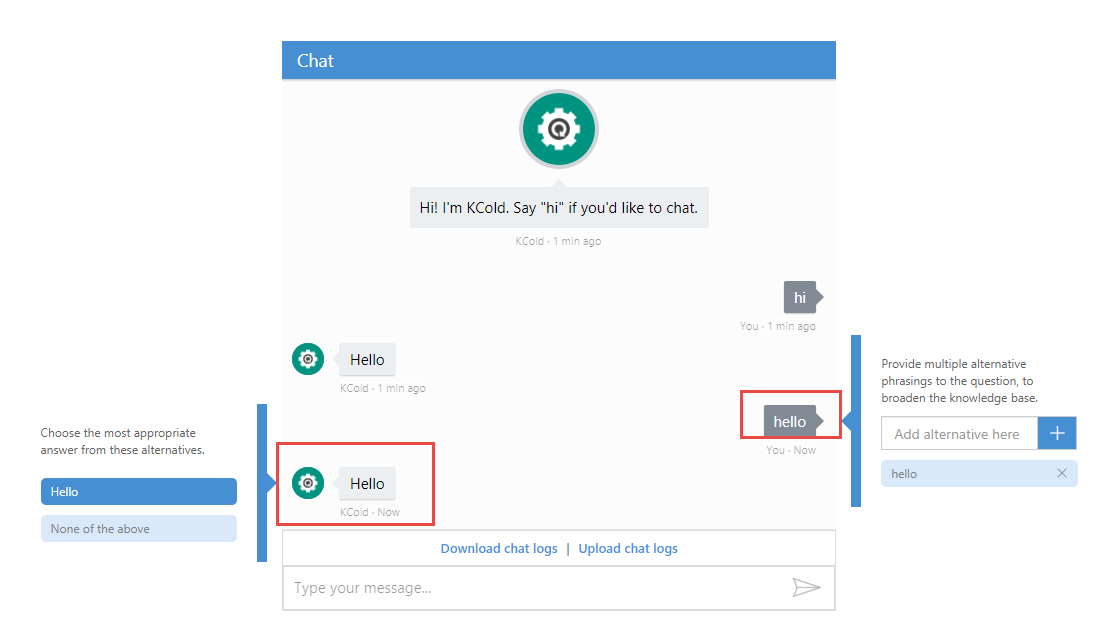
1. Add alternative to common words like **Hi** and **Hello**



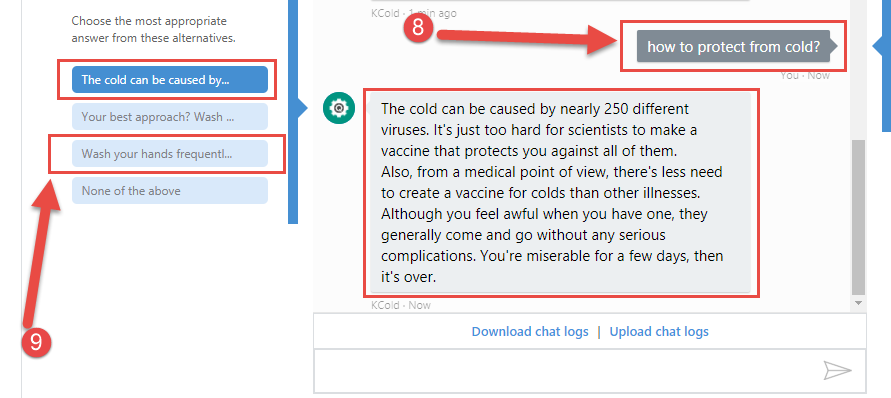
1. Test the modification, type **Hello**



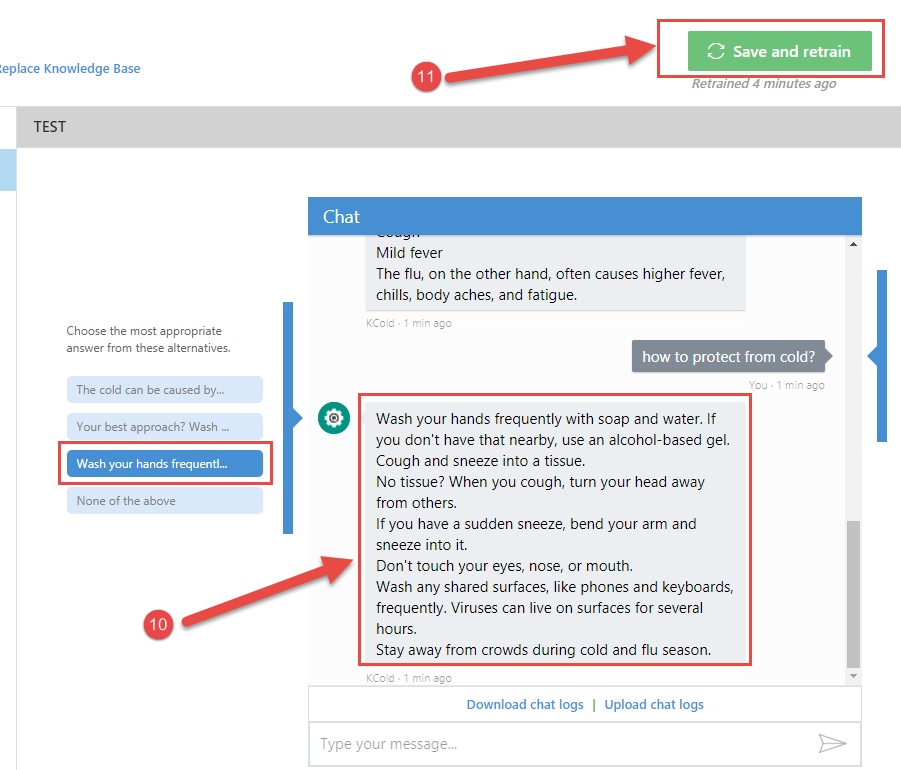
1. Answer provided:



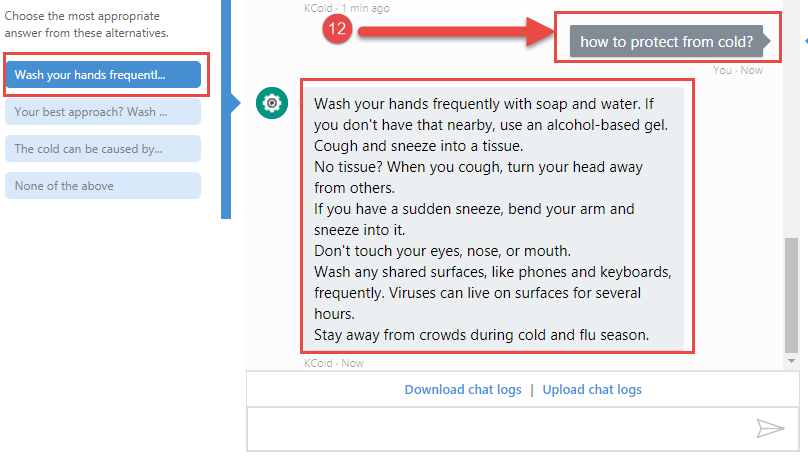
1. Ask questions in different ways and choose the most appropriate answer



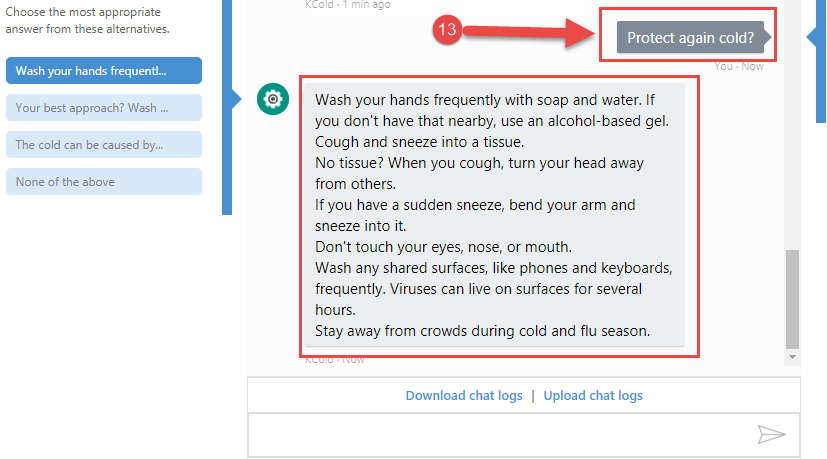
1. After selection the new response, click on **Save and retrain**



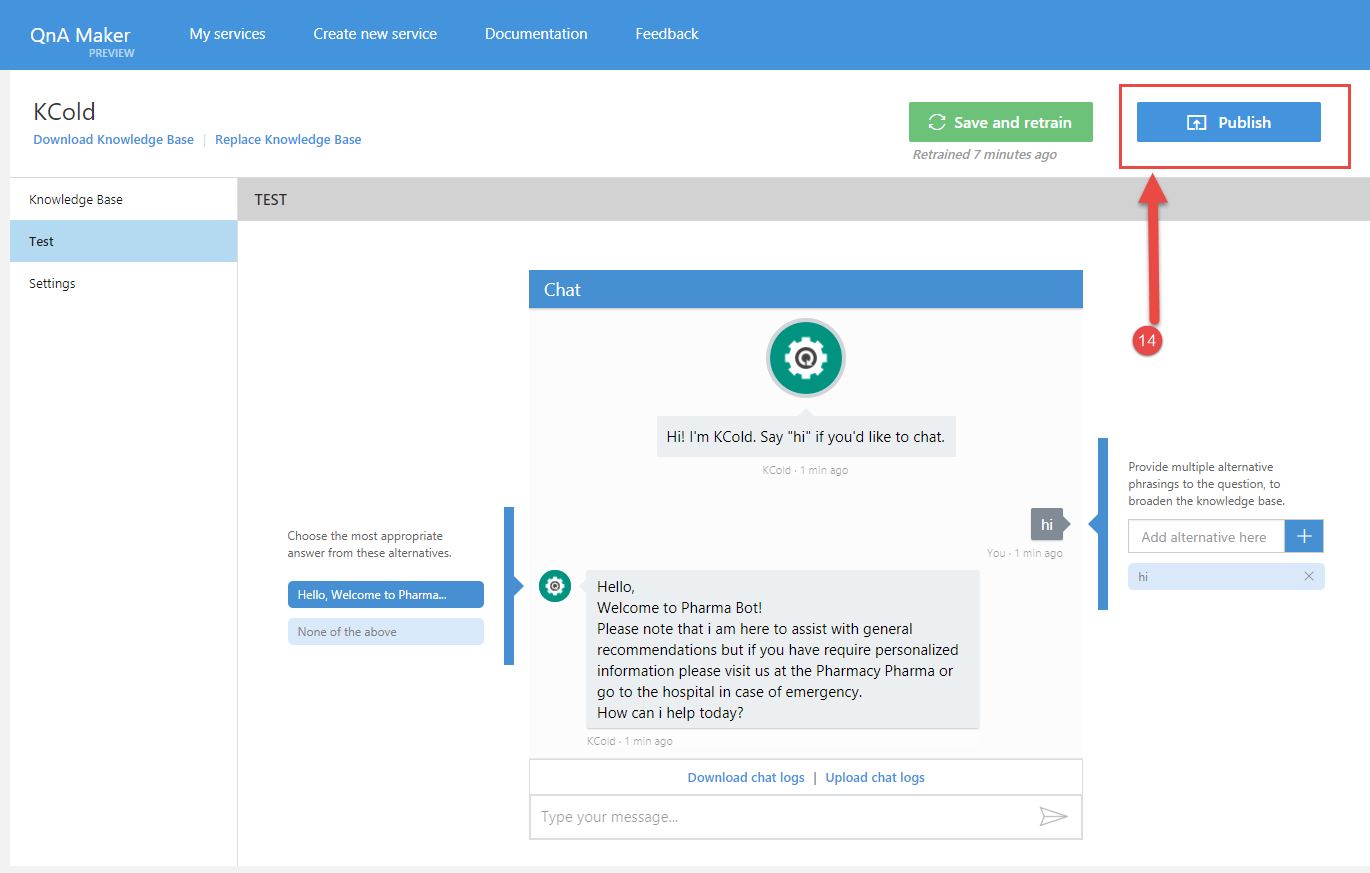
1. Test by asking the question again



1. Ask the same question using different word to test



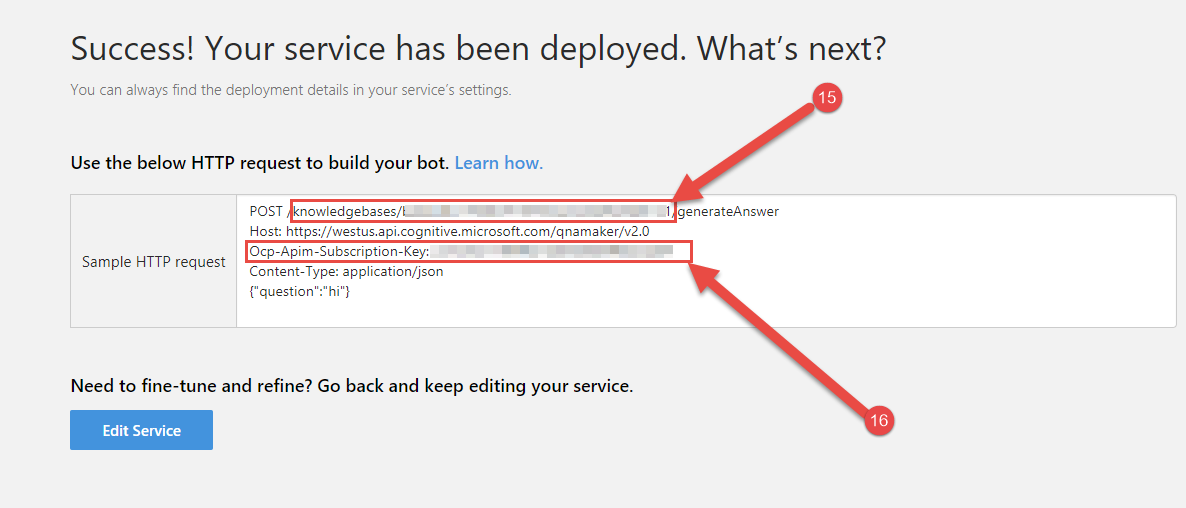
1. Repeat Steps 9 -12 for other questions to increase the performance of the AI so that it responds better to a variety of questions.
2. Once comfortable with the responses click on **publish**



1. Click on **Publish** again



1. At the screen below, capture the two information below to configure the Web Bot once created.



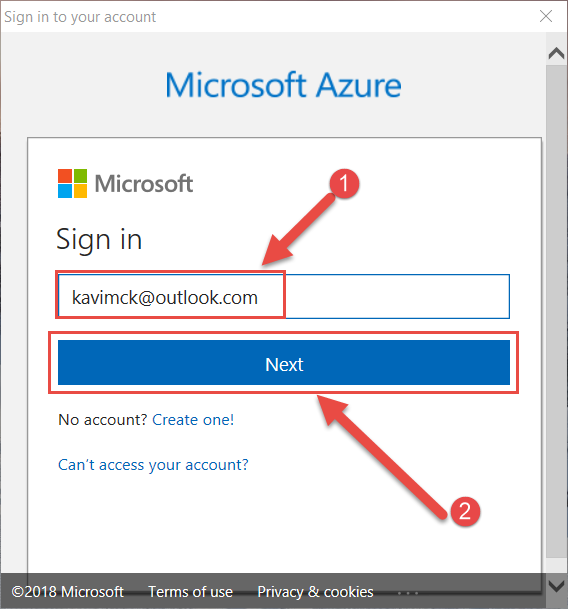
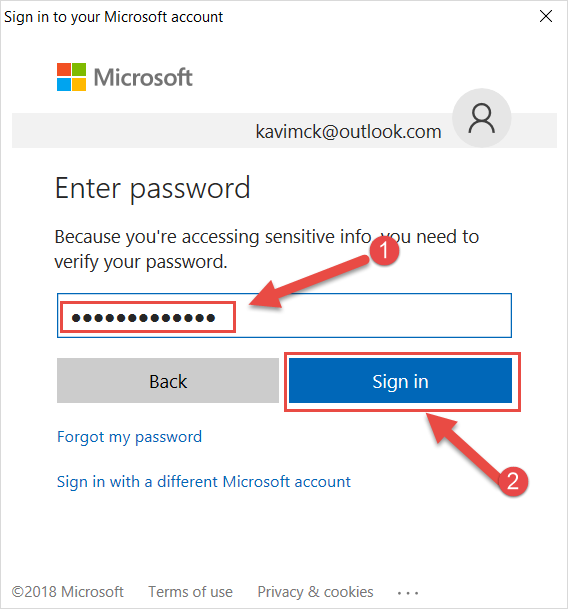
## Create Resource Group

1. Open PowerShell

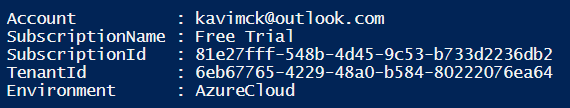
*Login-AzureRmAccount*



1. Login with Azure account

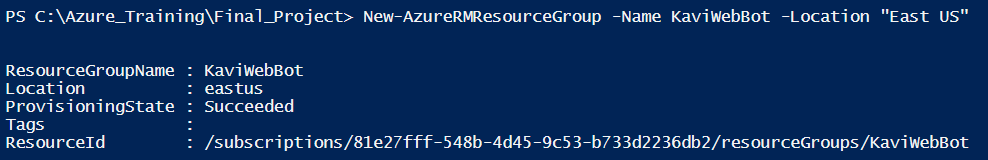


1. Confirmation that you are now logged in



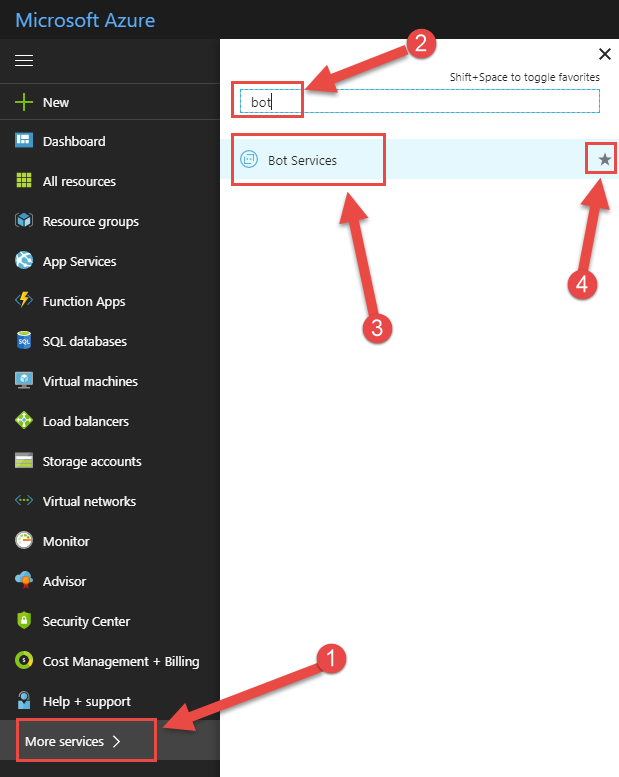
1. Create Resource Group

*New-AzureRMResourceGroup -Name KaviWebBot -Location "East US"*

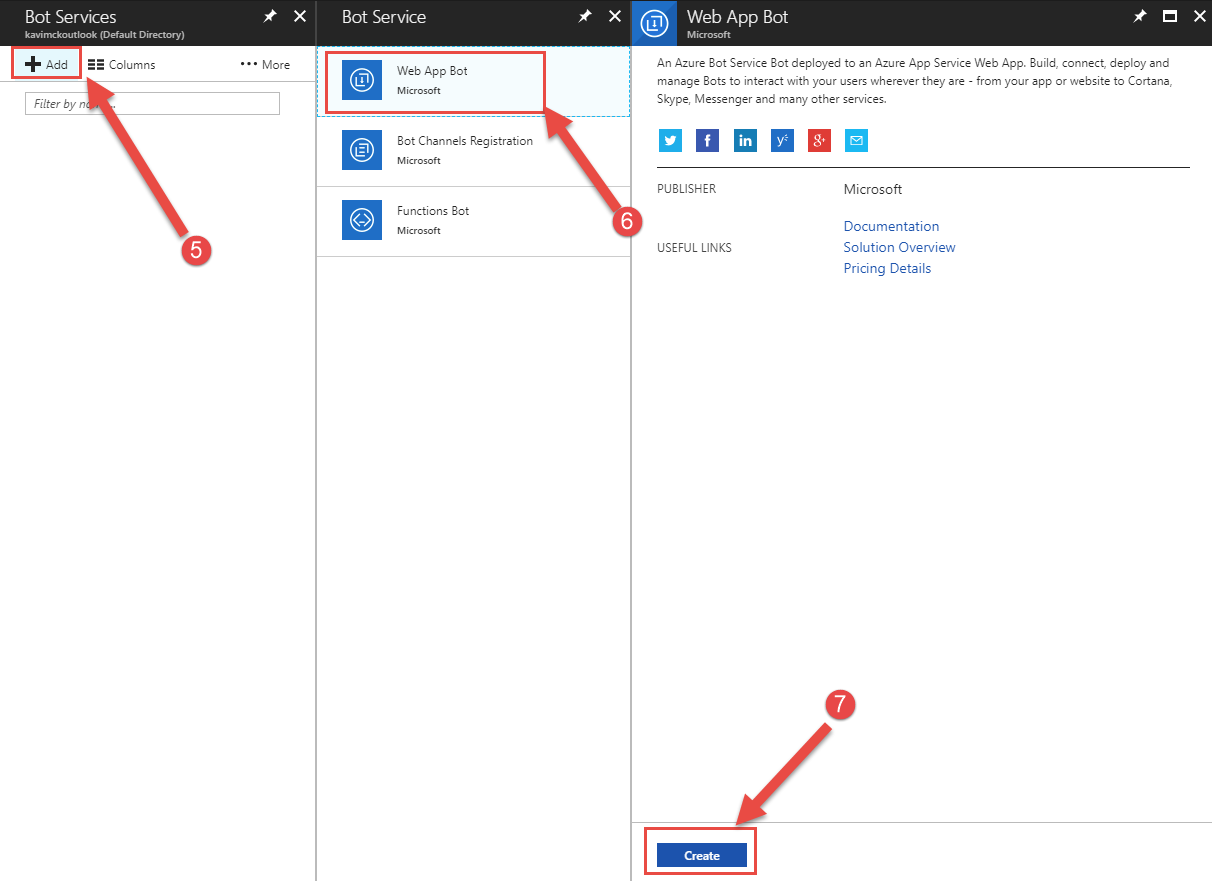


## Create Azure Web Bot

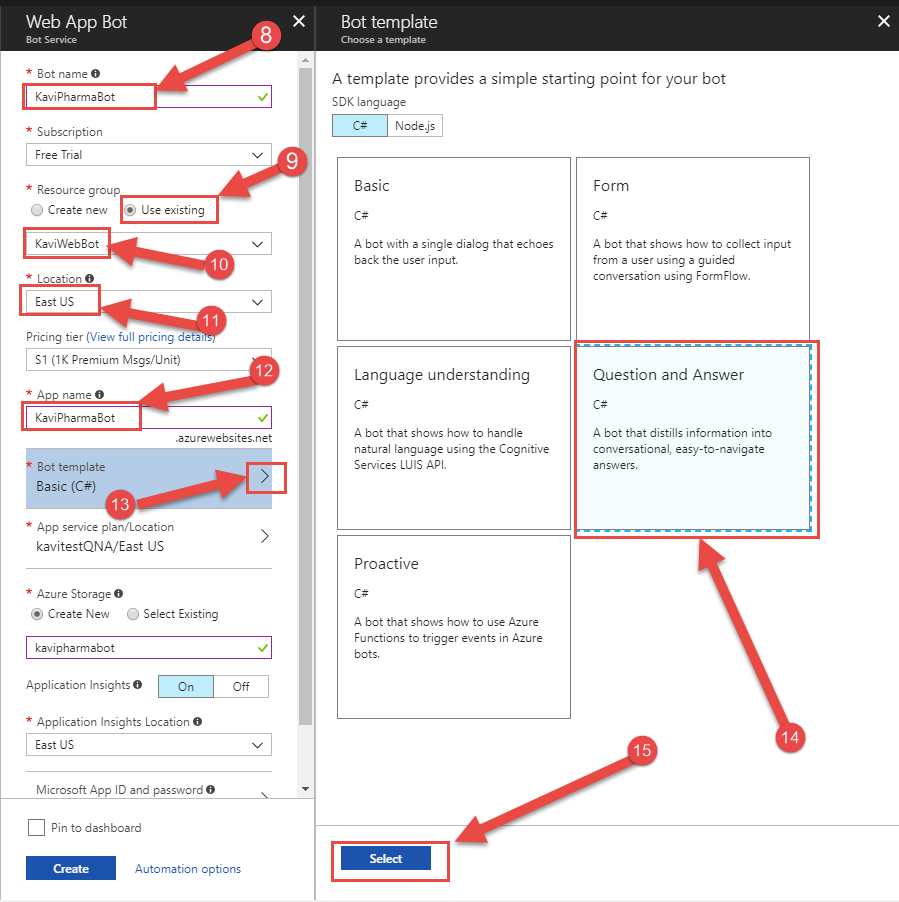
1. Login on the Azure Portal (<https://portal.azure.com>)
2. Follow the steps below to access the Bot Services menu



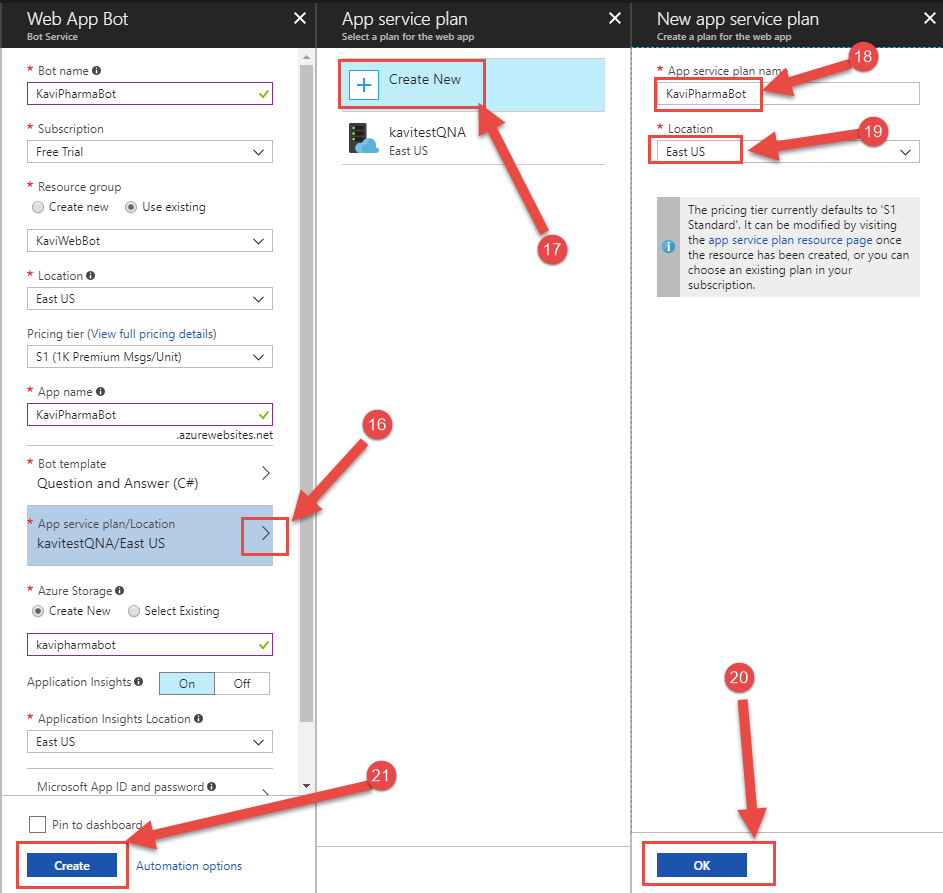
1. Click on **Add** and follow the steps to create a new Bot



1. Follow the steps below to select a **Question and Answer Bot Template**

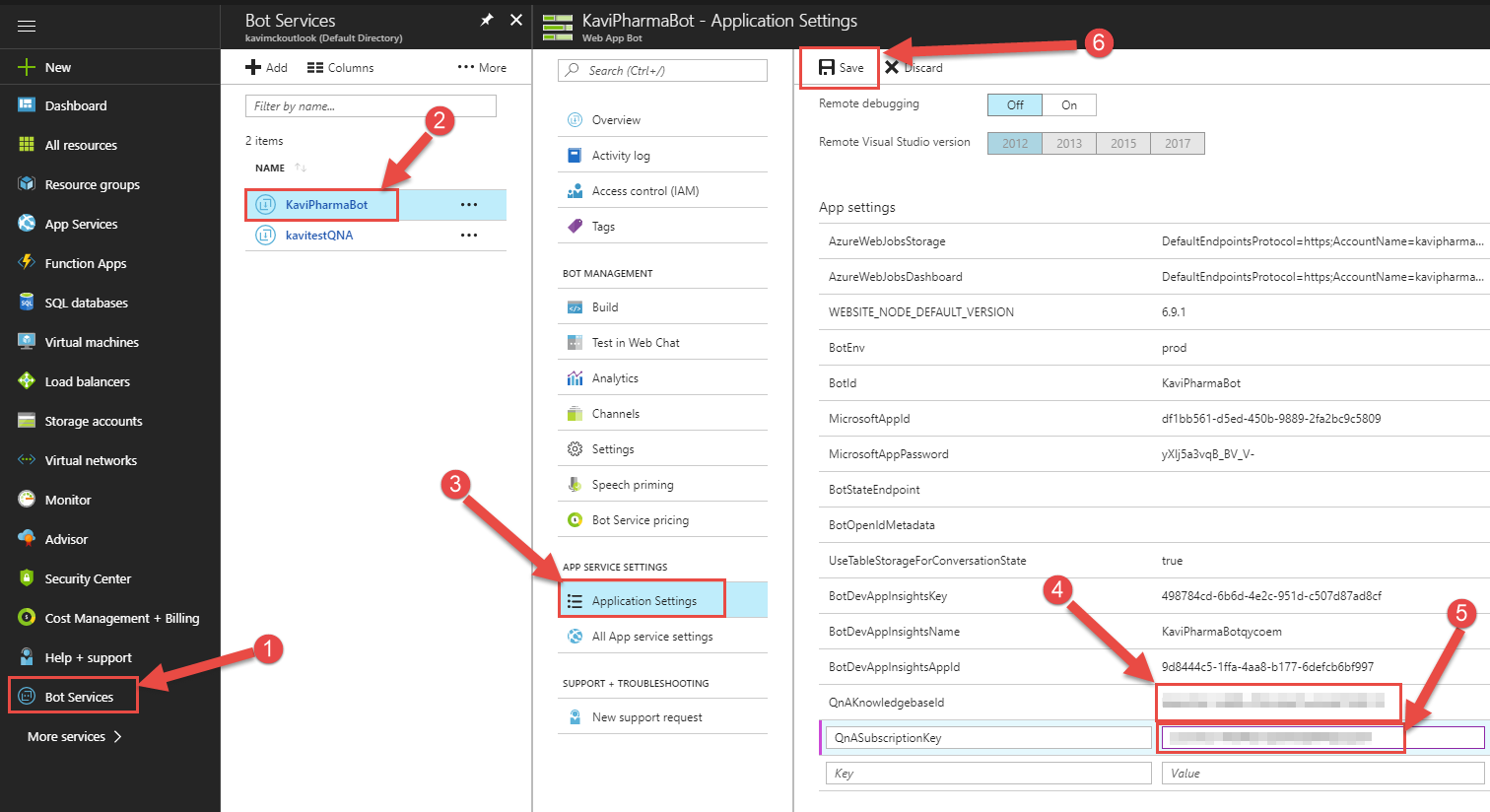


1. Continue with the steps below to create the Web Bot

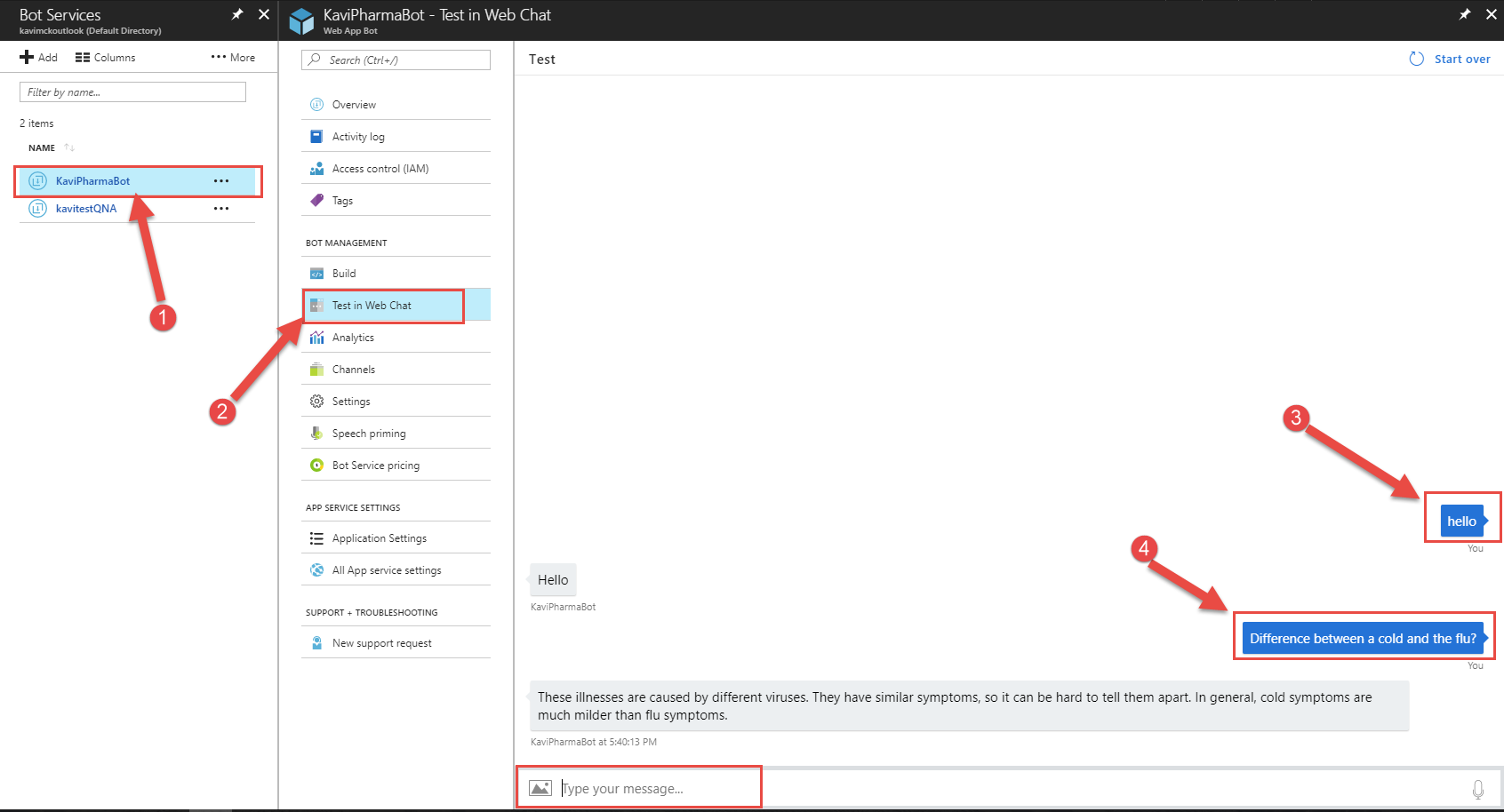


## Associate Web Bot with QnA Maker

1. Follow the steps below and enter the 2 keys captured from QnA Maker

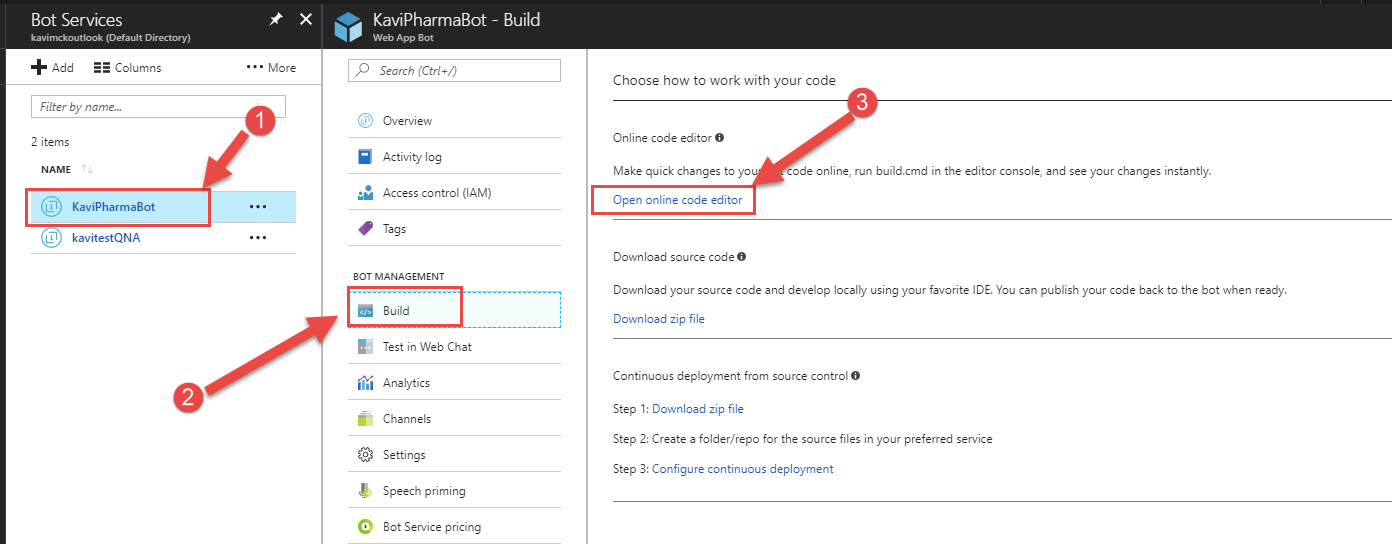


1. Test connection to the QnA by following the steps below

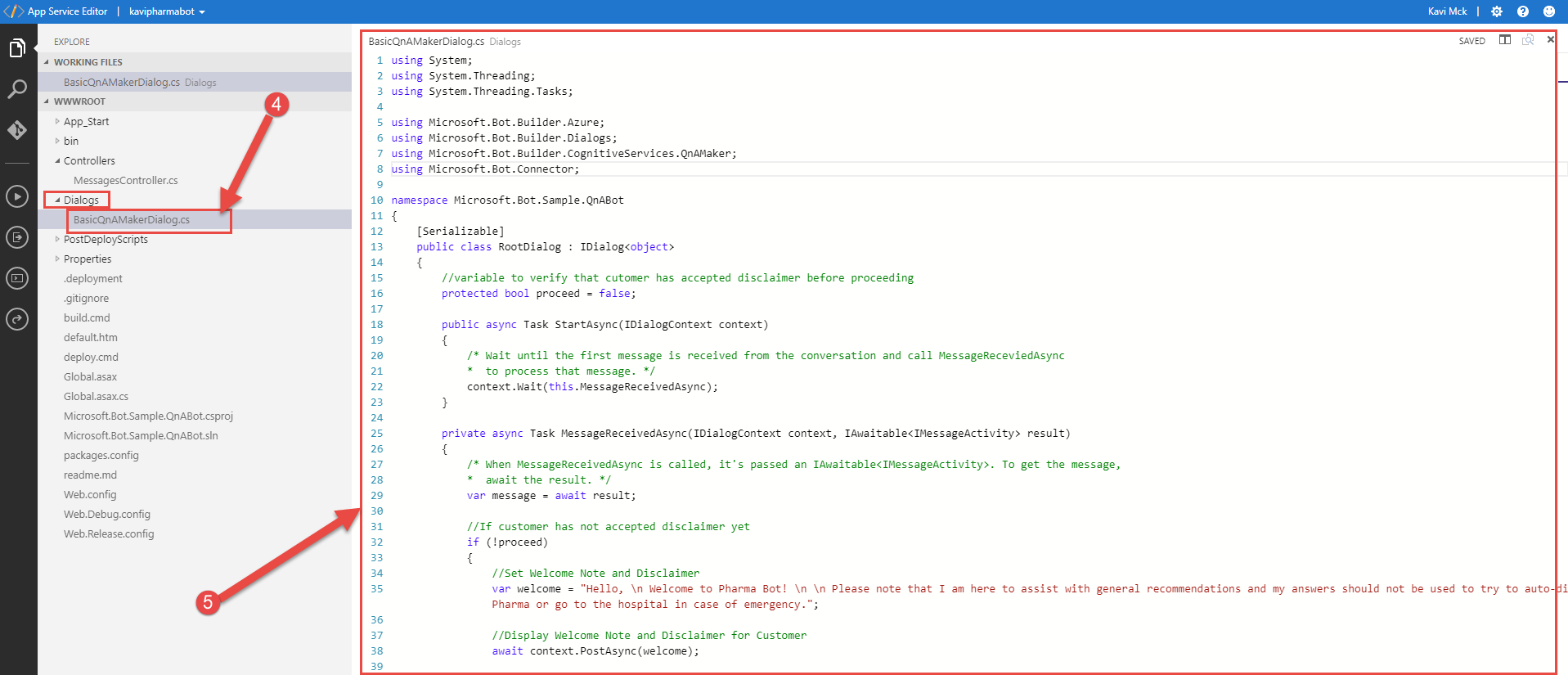


## Add a Disclaimer for the customers to accept

1. Follow the steps below to access the code of the Web Bot using the **Online code Editor**



1. Open the file **BasicQnAMakerDialog.cs**



1. Modify the code as follows

*using System;*

*using System.Threading;*

*using System.Threading.Tasks;*

*using Microsoft.Bot.Builder.Azure;*

*using Microsoft.Bot.Builder.Dialogs;*

*using Microsoft.Bot.Builder.CognitiveServices.QnAMaker;*

*using Microsoft.Bot.Connector;*

*namespace Microsoft.Bot.Sample.QnABot*

*{*

*[Serializable]*

*public class RootDialog : IDialog<object>*

*{*

*//variable to verify that cutomer has accepted disclaimer before proceeding*

*protected bool proceed = false;*

*public async Task StartAsync(IDialogContext context)*

*{*

*/\* Wait until the first message is received from the conversation and call MessageReceviedAsync*

*\* to process that message. \*/*

*context.Wait(this.MessageReceivedAsync);*

*}*

*private async Task MessageReceivedAsync(IDialogContext context, IAwaitable<IMessageActivity> result)*

*{*

*/\* When MessageReceivedAsync is called, it's passed an IAwaitable<IMessageActivity>. To get the message,*

*\* await the result. \*/*

*var message = await result;*

*//If customer has not accepted disclaimer yet*

*if (!proceed)*

*{*

*//Set Welcome Note and Disclaimer*

*var welcome = "Hello, \n Welcome to Pharma Bot! \n \n Please note that I am here to assist with general recommendations and my answers should not be used to try to auto-diagnose your health.\n \n If you have have personalized questions or concerns please visit us at the Pharmacy Pharma or go to the hospital in case of emergency.";*

*//Display Welcome Note and Disclaimer for Customer*

*await context.PostAsync(welcome);*

*//Ask the Custmer to accept Disclaimer by calling function CustomerInfo*

*PromptDialog.Confirm(*

*context,*

*CustomerInfo,*

*"Do you want to proceed?",*

*"Didn't get that!",*

*promptStyle: PromptStyle.Auto);*

*}*

*//capture the information that we entered from the QnA maker*

*var qnaSubscriptionKey = Utils.GetAppSetting("QnASubscriptionKey");*

*var qnaKBId = Utils.GetAppSetting("QnAKnowledgebaseId");*

*//if proceed is true (Customer accepted the disclaimer)*

*if (proceed)*

*{*

*//if Customer says bye during conversation*

*if (message.Text == "Bye" || message.Text == "bye" || message.Text == "BYE")*

*{*

*//set proceed to false so that customer goes through disclaimer again*

*this.proceed = false;*

*//respond by bye*

*await context.PostAsync("Hope i was able to answer your questions \n Good Bye!");*

*}*

*else*

*{*

*// Validate if QnA Subscription Key and KnowledgeBase Id are not null*

*if (!string.IsNullOrEmpty(qnaSubscriptionKey) && !string.IsNullOrEmpty(qnaKBId))*

*{*

*await context.Forward(new BasicQnAMakerDialog(), AfterAnswerAsync, message, CancellationToken.None);*

*}*

*// if either the QnA Subscription Key or KnowledgeBase Id is not set*

*else*

*{*

*await context.PostAsync("Please set QnAKnowledgebaseId and QnASubscriptionKey in App Settings. Get them at https://qnamaker.ai.");*

*}*

*}*

*}*

*}*

*private async Task AfterAnswerAsync(IDialogContext context, IAwaitable<IMessageActivity> result)*

*{*

*// wait for the next user message*

*context.Wait(MessageReceivedAsync);*

*}*

*//Validates if customer has accepted disclaimer*

*public async Task CustomerInfo(IDialogContext context, IAwaitable<bool> argument)*

*{*

*//capture the customer's answer to the disclaimer*

*var confirm = await argument;*

*//If answer is YES*

*if (confirm)*

*{*

*//modify the boolean variable proceed to true*

*this.proceed = true;*

*//Ask the customer what he would like to know about the COLD*

*await context.PostAsync("Great!\n What would you like to learn about the common COLD or FLU today?");*

*}*

*//If answer is NO*

*else*

*{*

*//say Bye to the customer*

*await context.PostAsync("Sorry i could not help you today :( \n Good Bye.\n See you next time!");*

*}*

*// wait for the next user message*

*context.Wait(MessageReceivedAsync);*

*}*

*}*

*// For more information about this template visit http://aka.ms/azurebots-csharp-qnamaker*

*[Serializable]*

*public class BasicQnAMakerDialog : QnAMakerDialog*

*{*

*// Go to https://qnamaker.ai and feed data, train & publish your QnA Knowledgebase.*

*// Parameters to QnAMakerService are:*

*// Required: subscriptionKey, knowledgebaseId,*

*// Optional: defaultMessage, scoreThreshold[Range 0.0 – 1.0]*

*public BasicQnAMakerDialog() : base(new QnAMakerService(new QnAMakerAttribute(Utils.GetAppSetting("QnASubscriptionKey"), Utils.GetAppSetting("QnAKnowledgebaseId"), "Sorry I don't have an answer for your question today. \n Please call us or visit us at the Pharmacy and one of my human colleagues will be happy to answer your question.", 0.5)))*

*{ }*

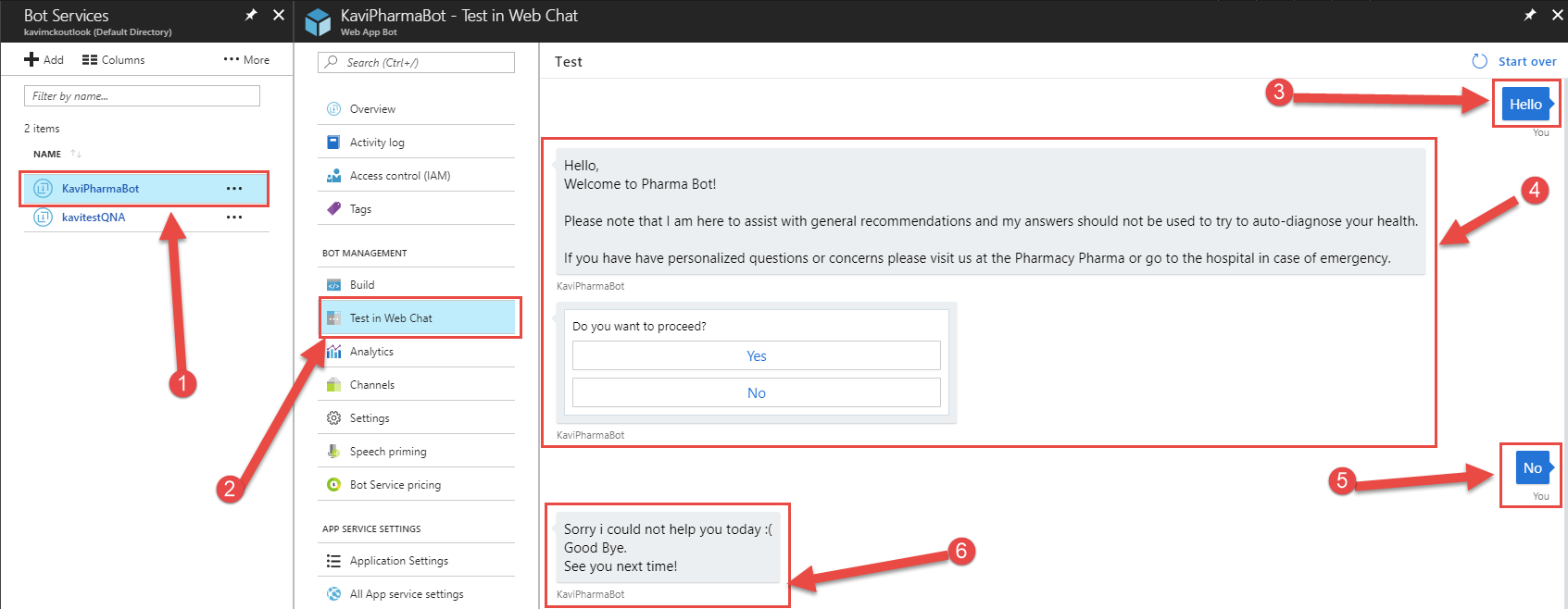
*}*

*}*

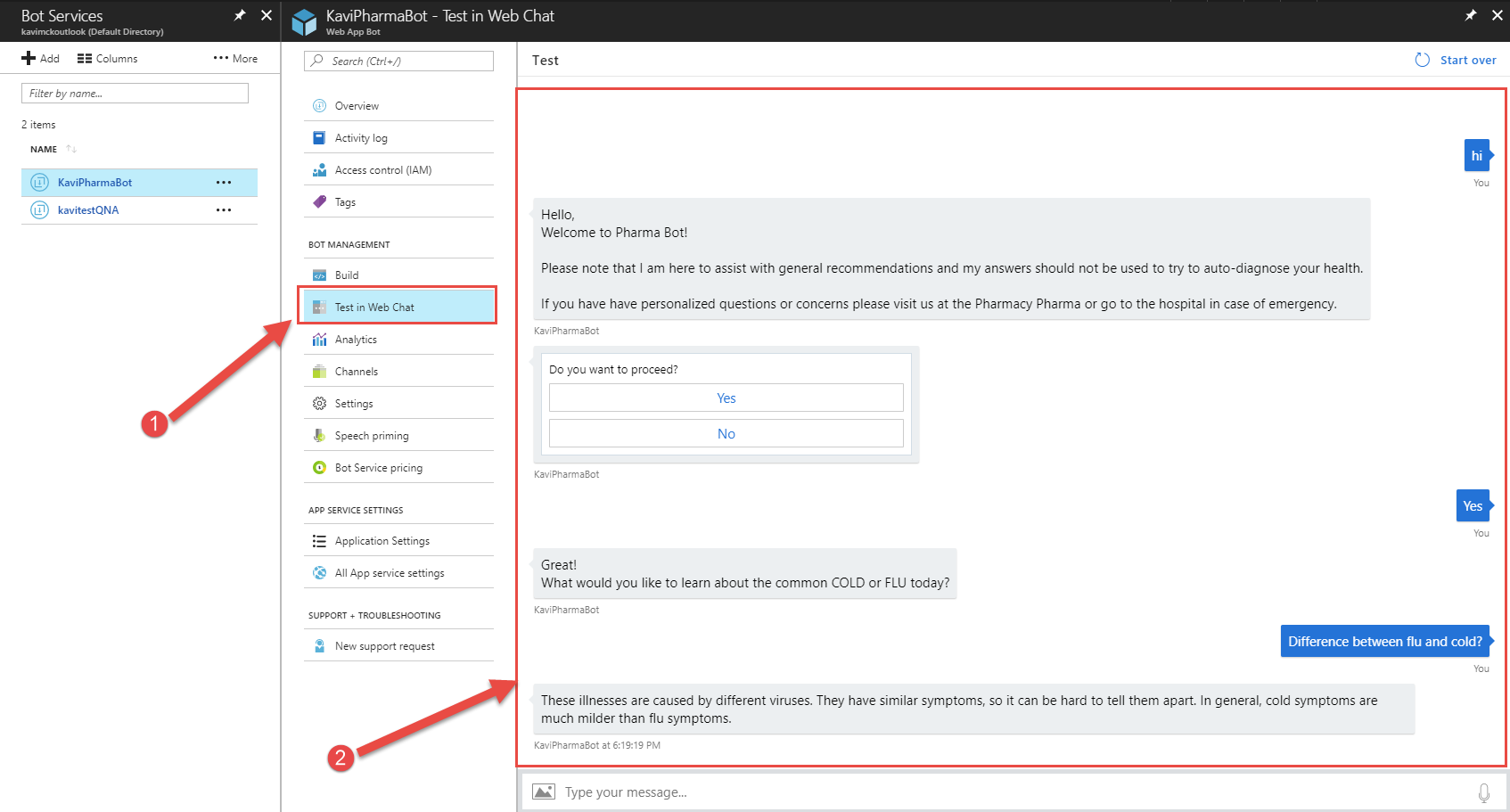
1. Apply the changes by Re-Building the project as follows



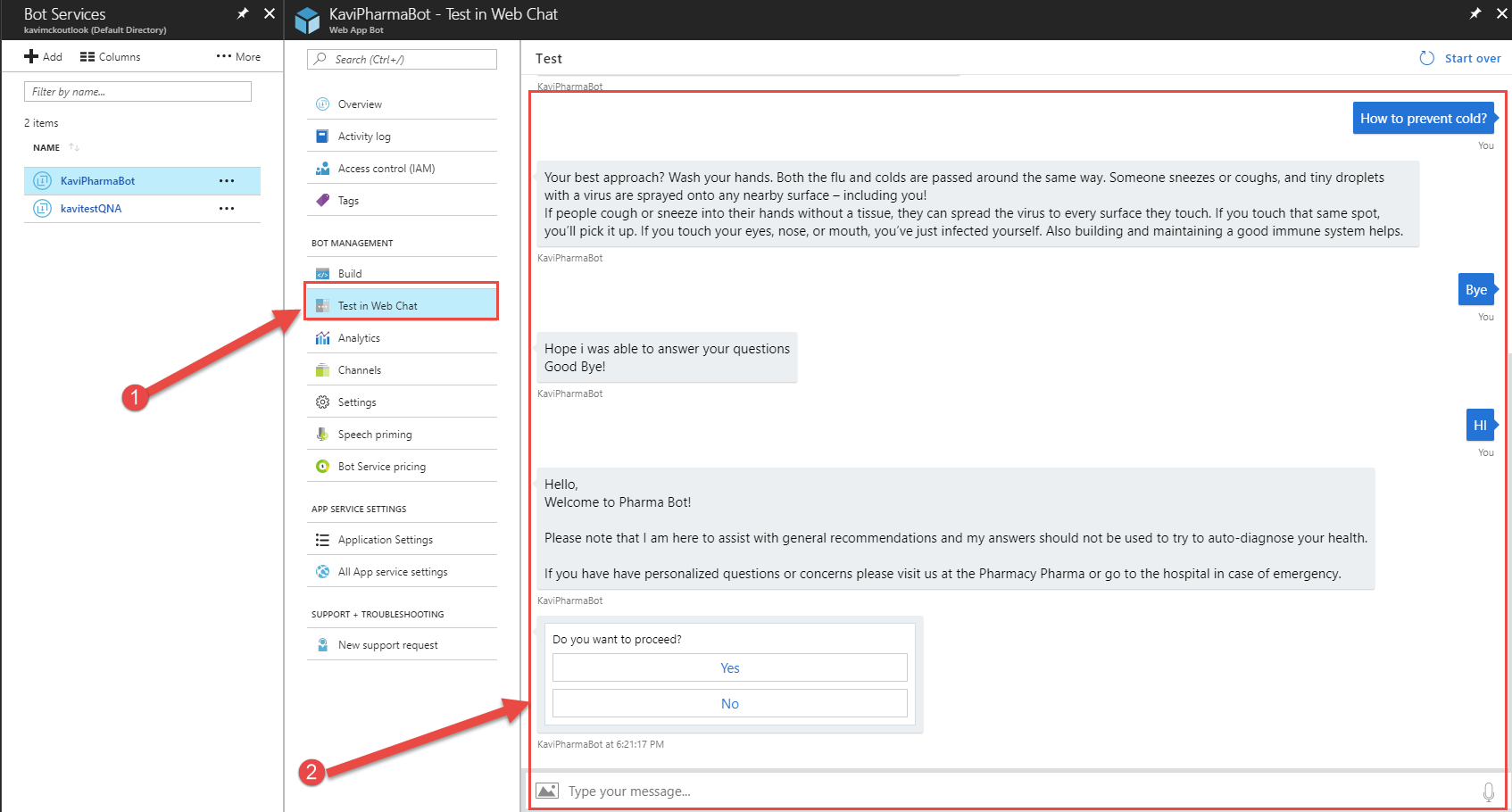
1. Test the Web Bot
   * Scenario: Disclaimer NOT accepted



* + Scenario: Disclaimer accepted

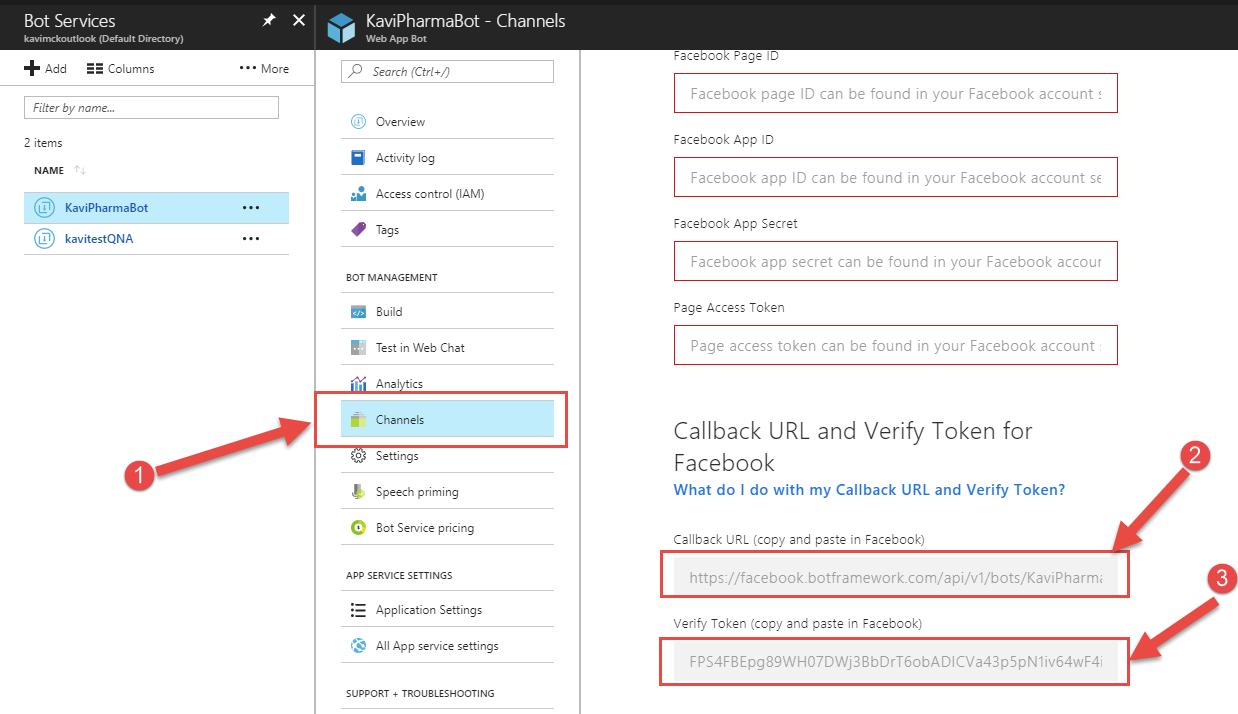


* + Scenario: Customer says Bye

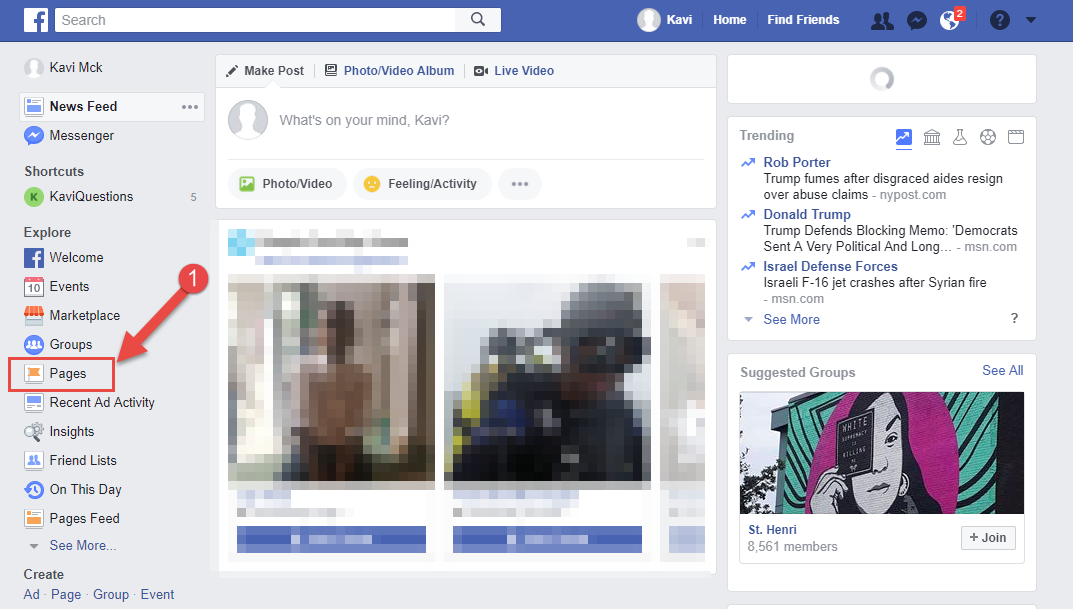


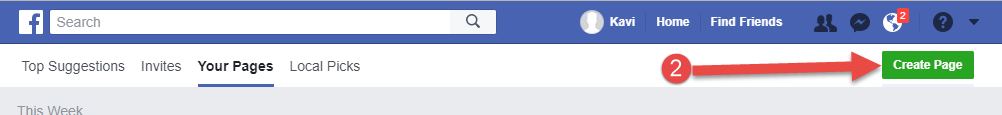
## Connecting to Facebook Messenger

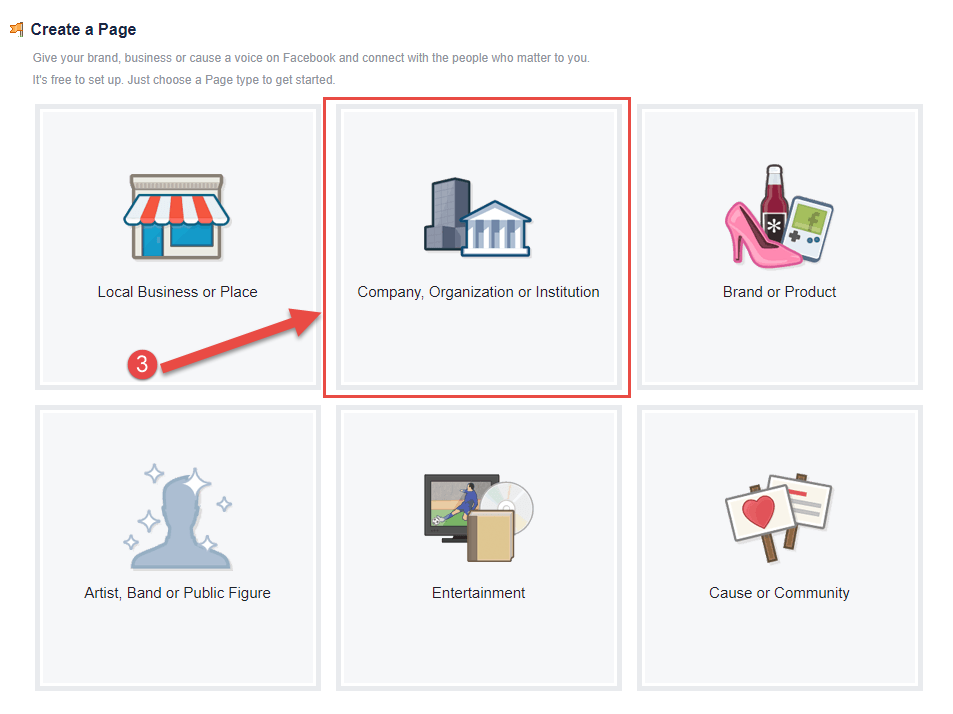
1. On Azure Portal, Click on Channels under the bot and capture the information for the Facebook app that we will create in the next step

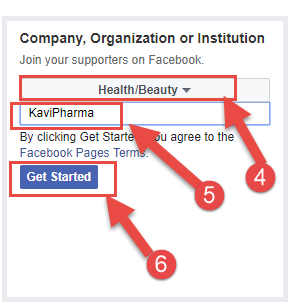


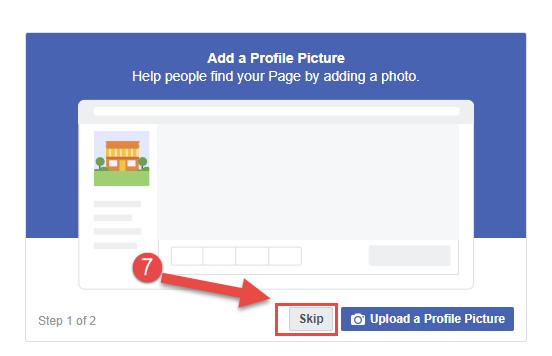
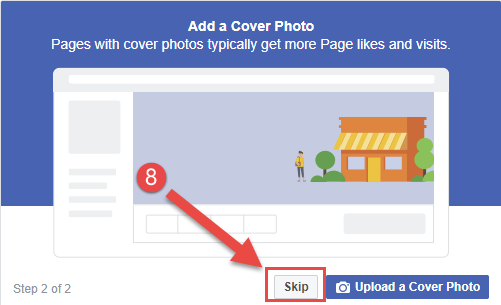
1. Login on Facebook and follow the steps below to create a new page ([www.facebook.com](http://www.facebook.com))







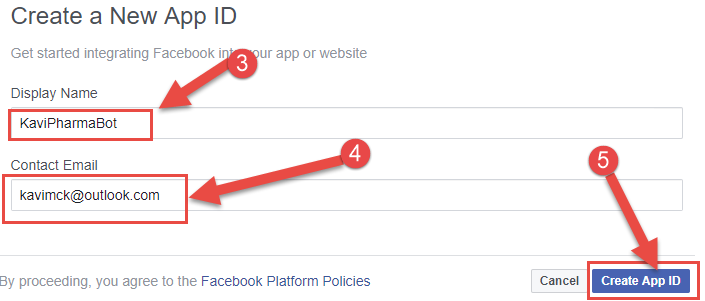


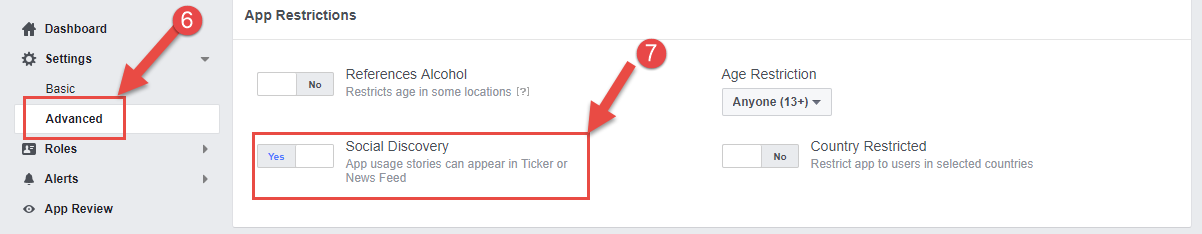


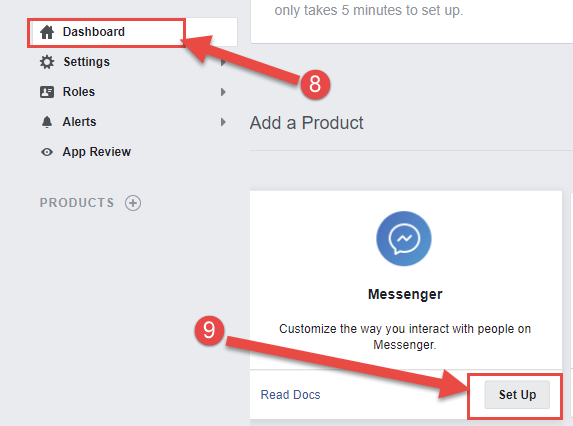


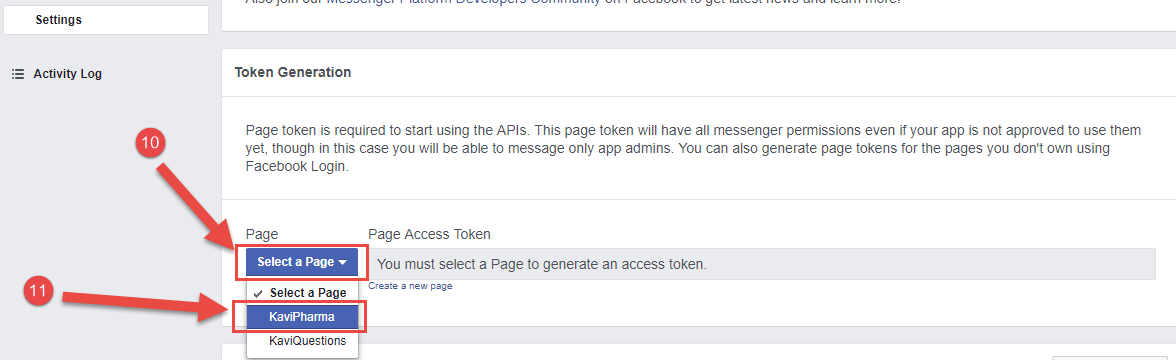
1. Create a Facebook App using your Facebook account by following the steps below (<https://developers.facebook.com/quickstarts/?platform=web>)

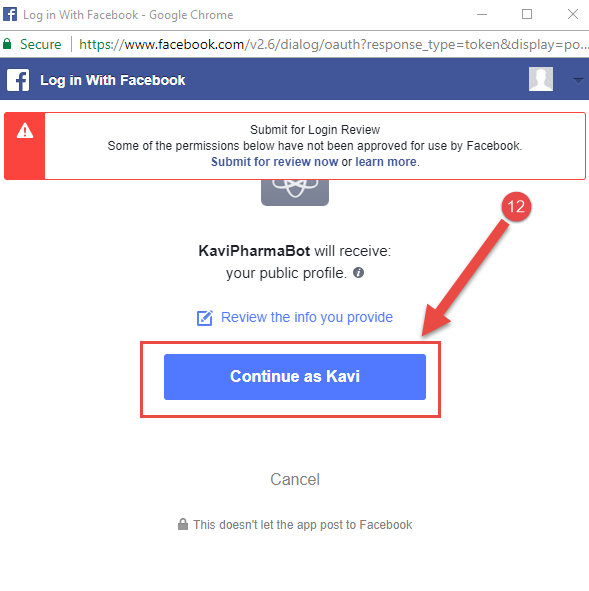


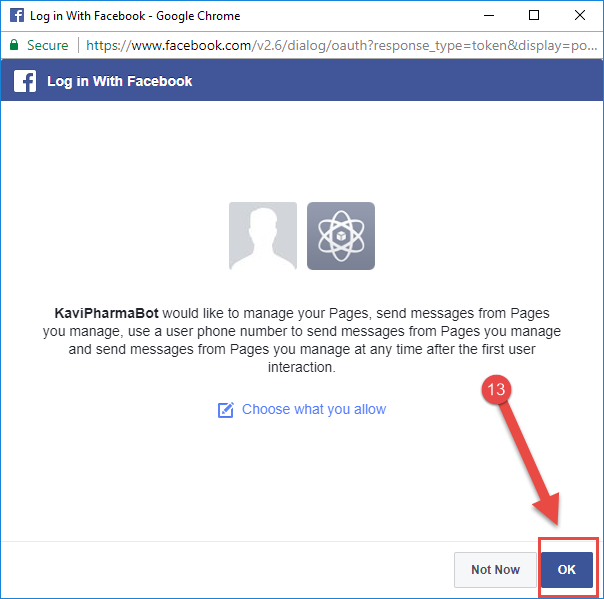


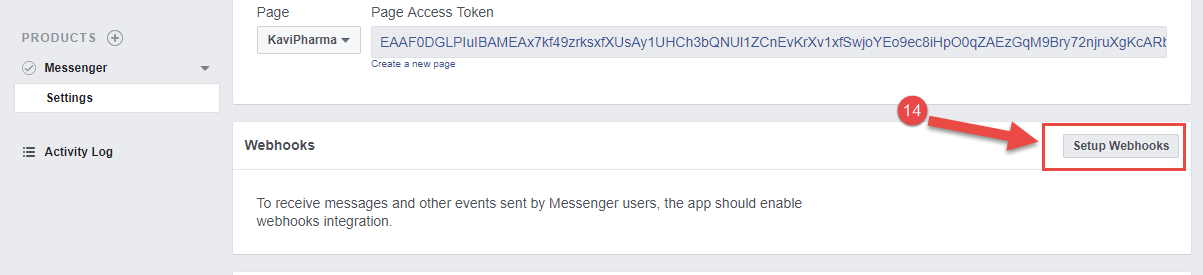


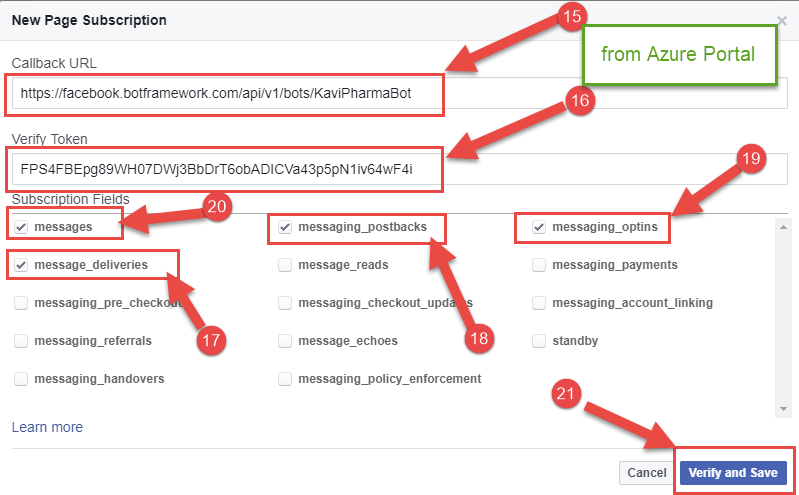


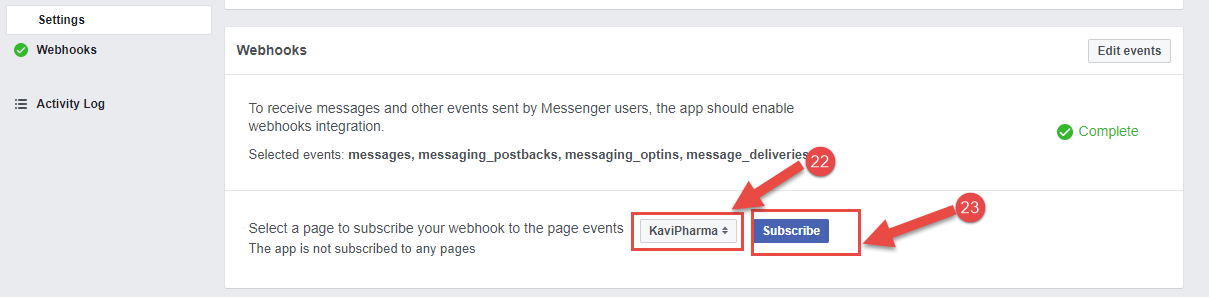


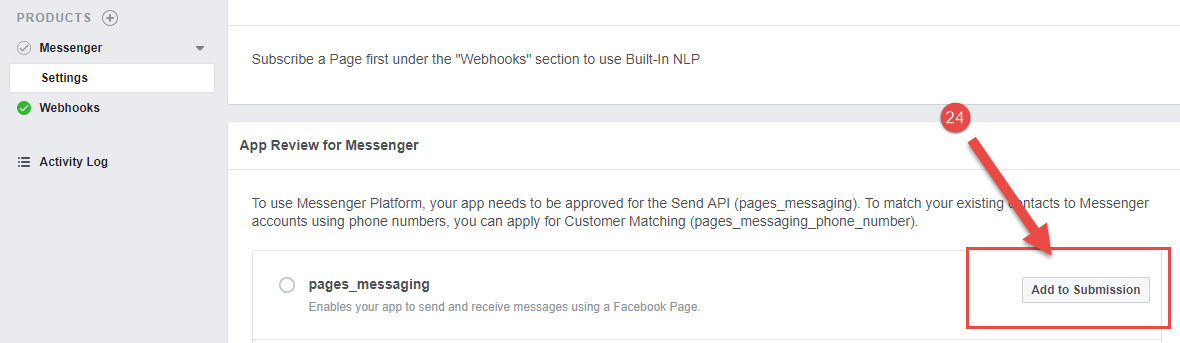


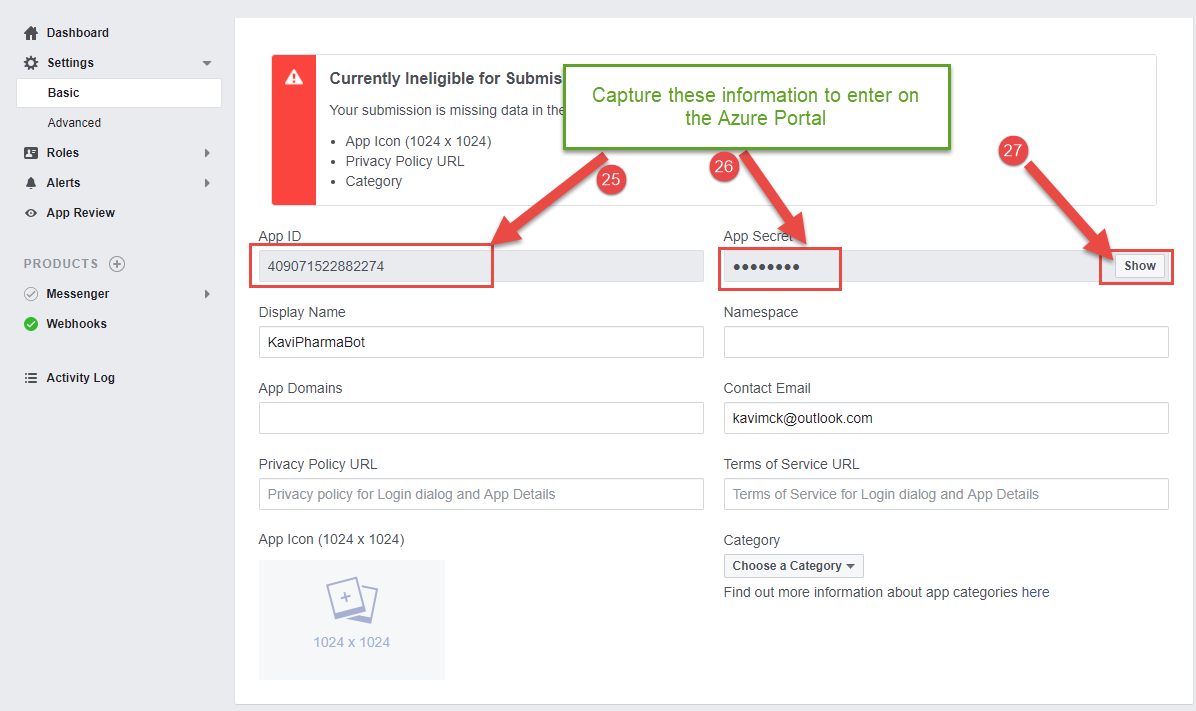


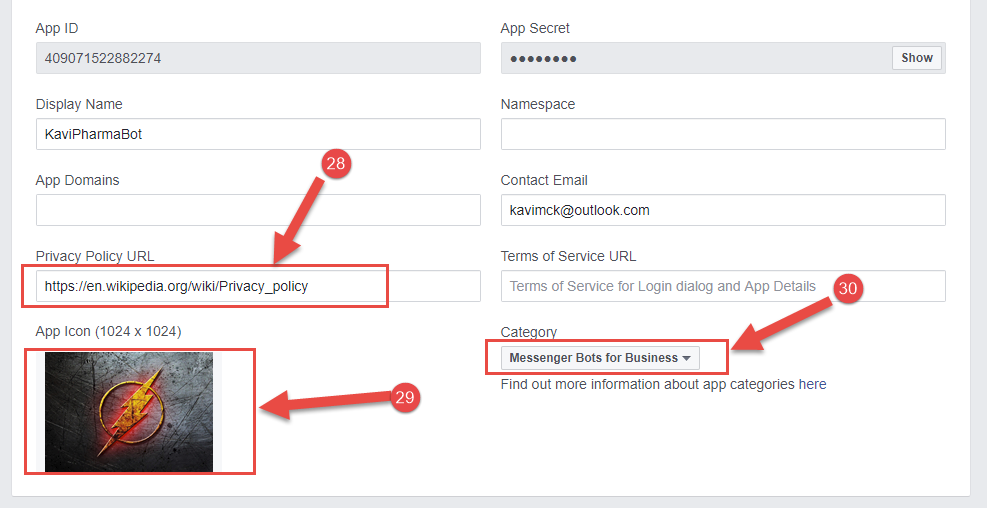




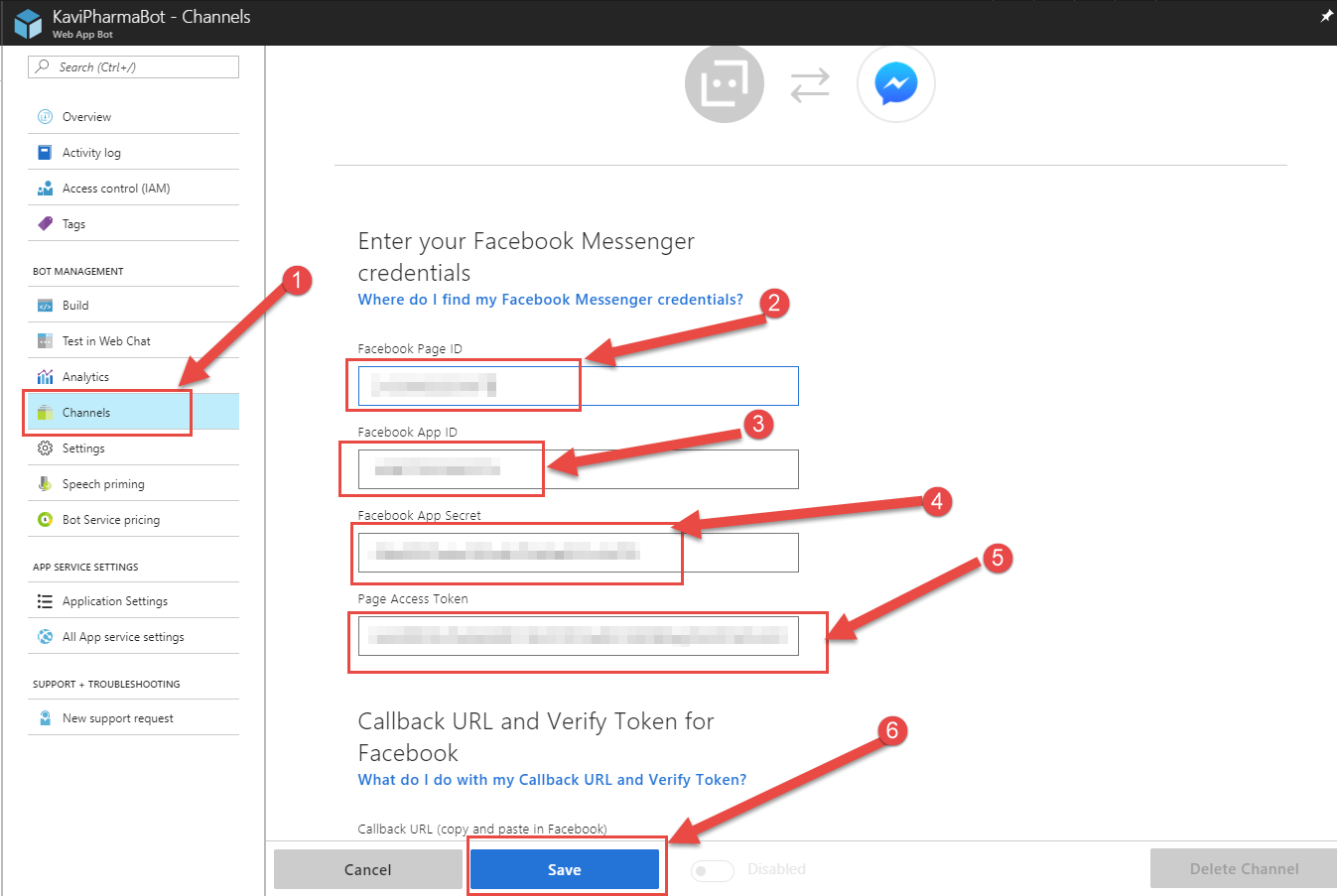


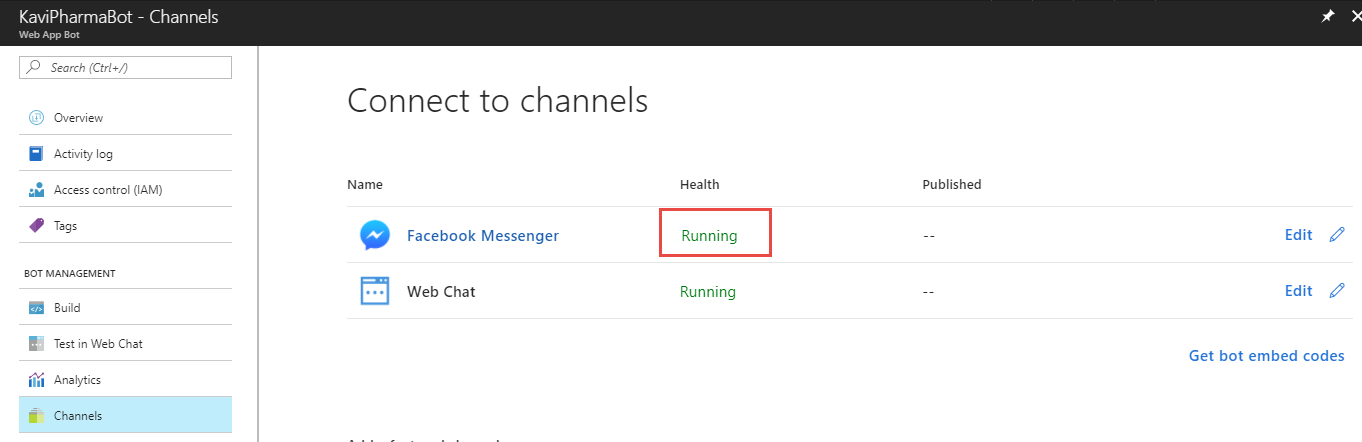




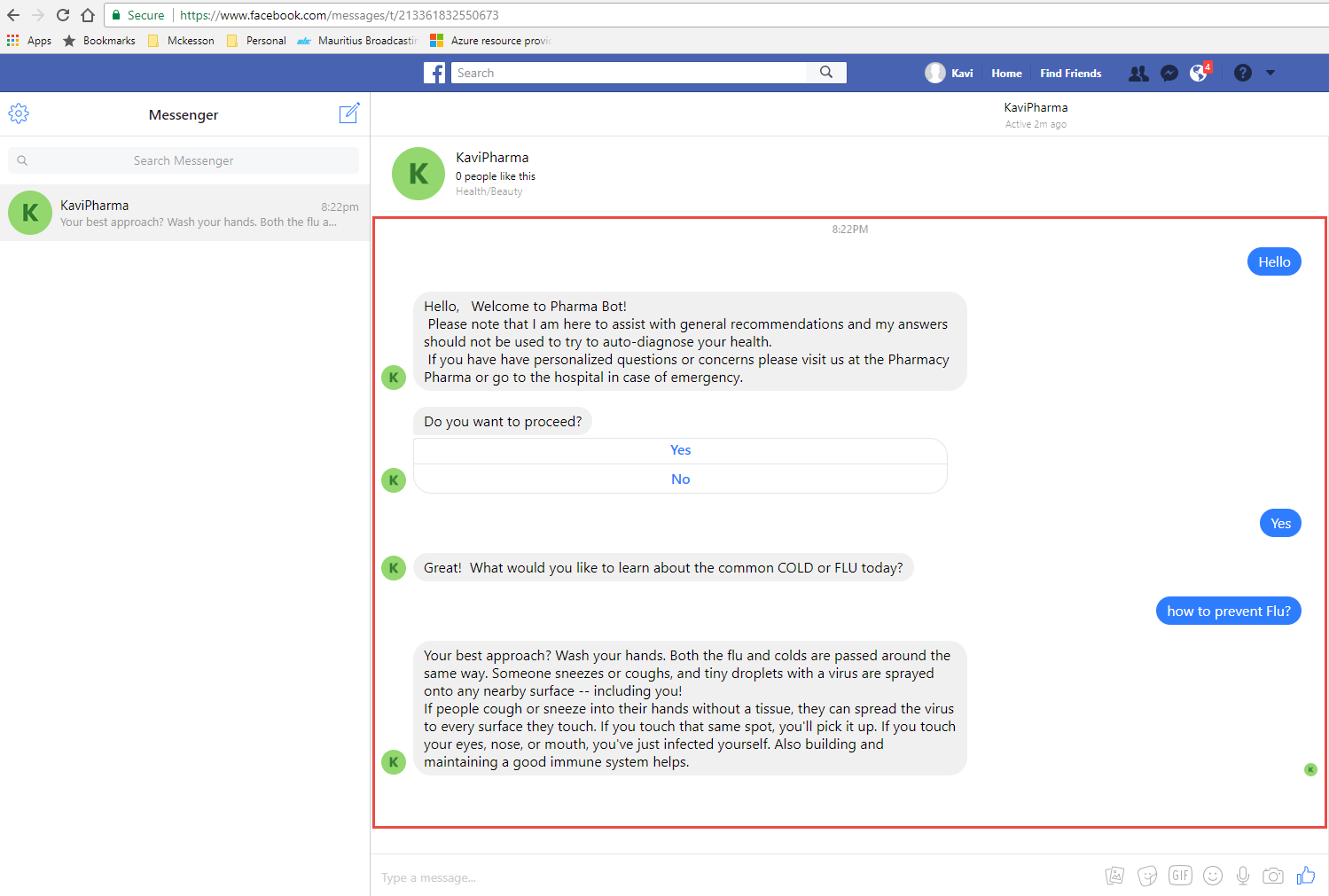


1. Go back to Azure Portal, enter the captured information from Facebook



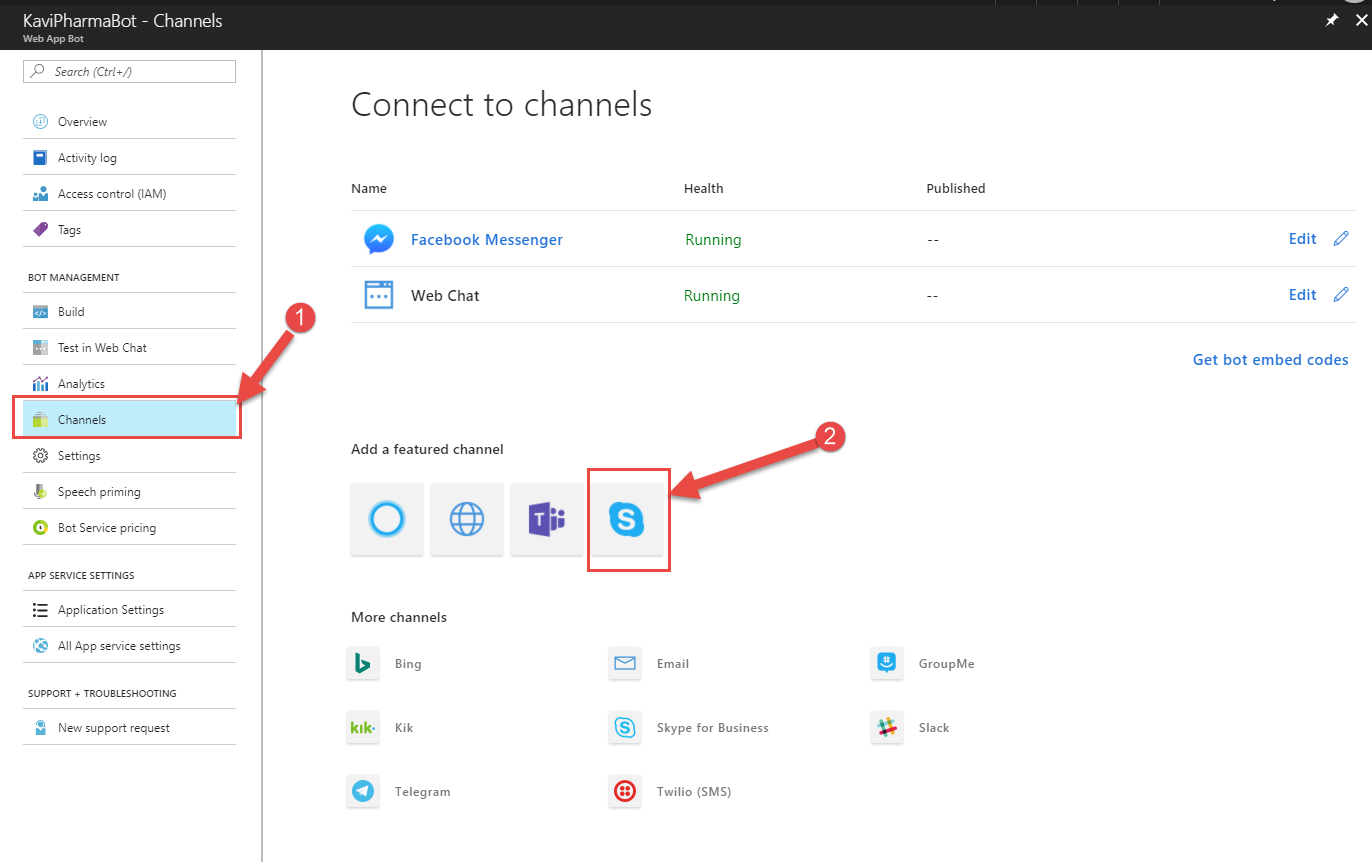


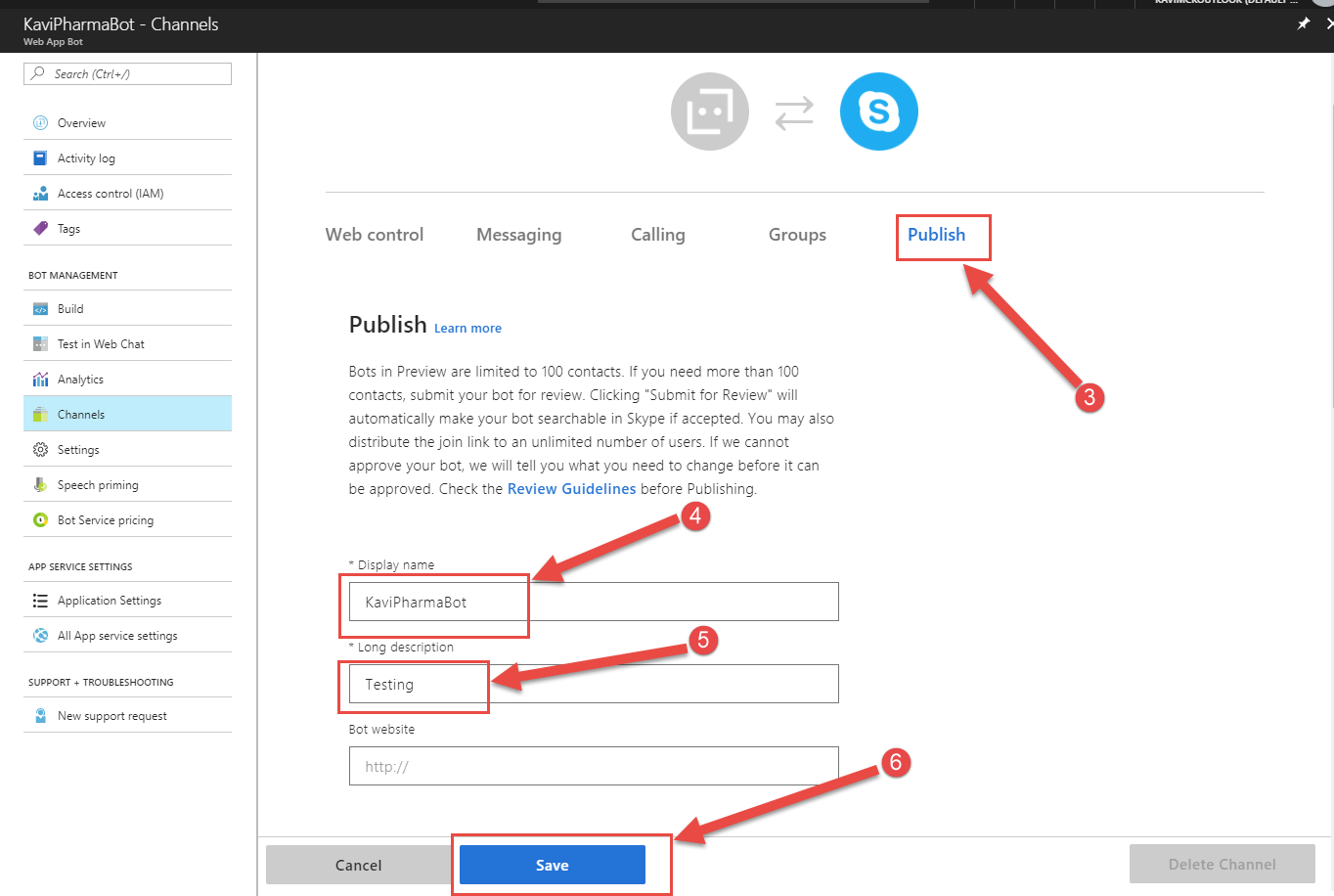
1. Open Messenger on Facebook page, search for **KaviPharma** and send some messages

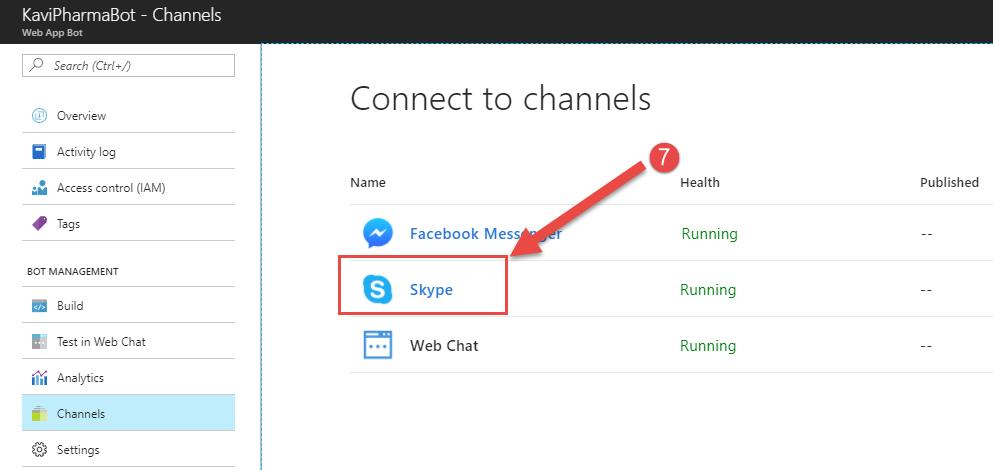


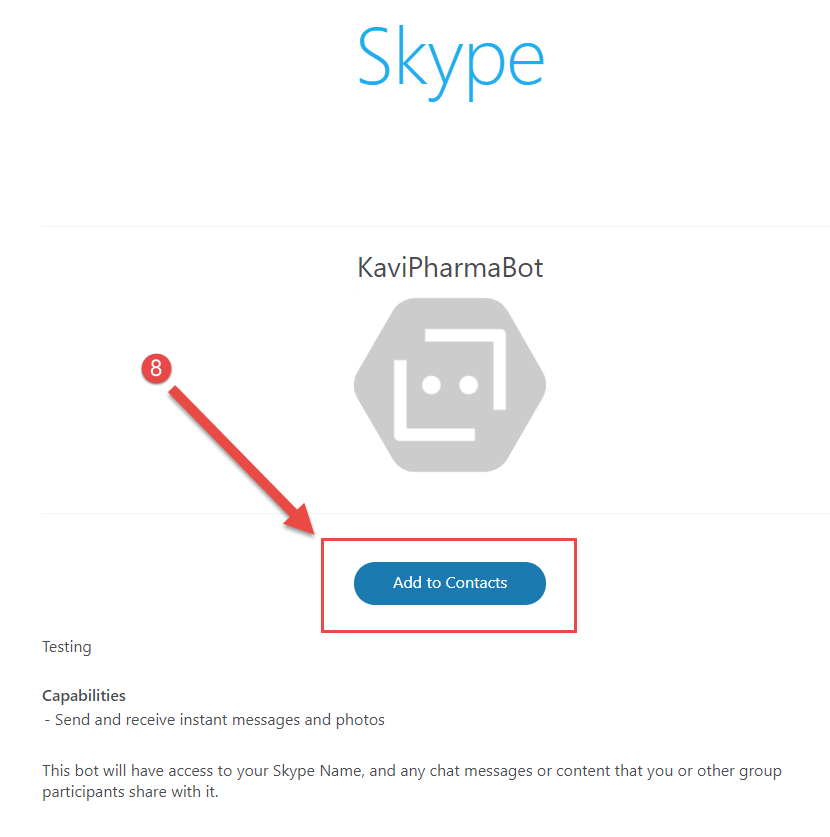
## Connecting to Skype

1. On Azure Portal, follow the steps below

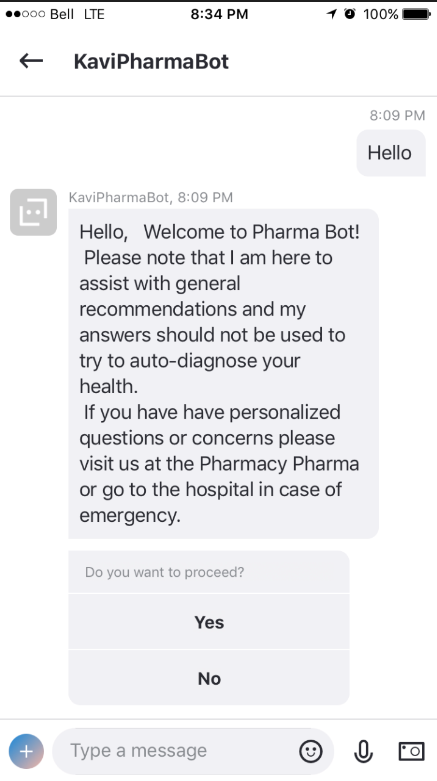
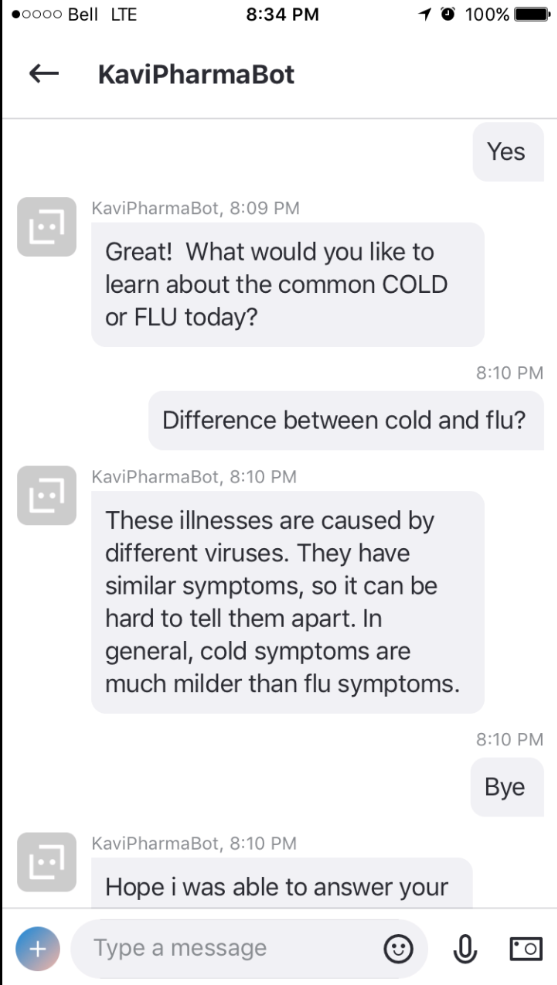






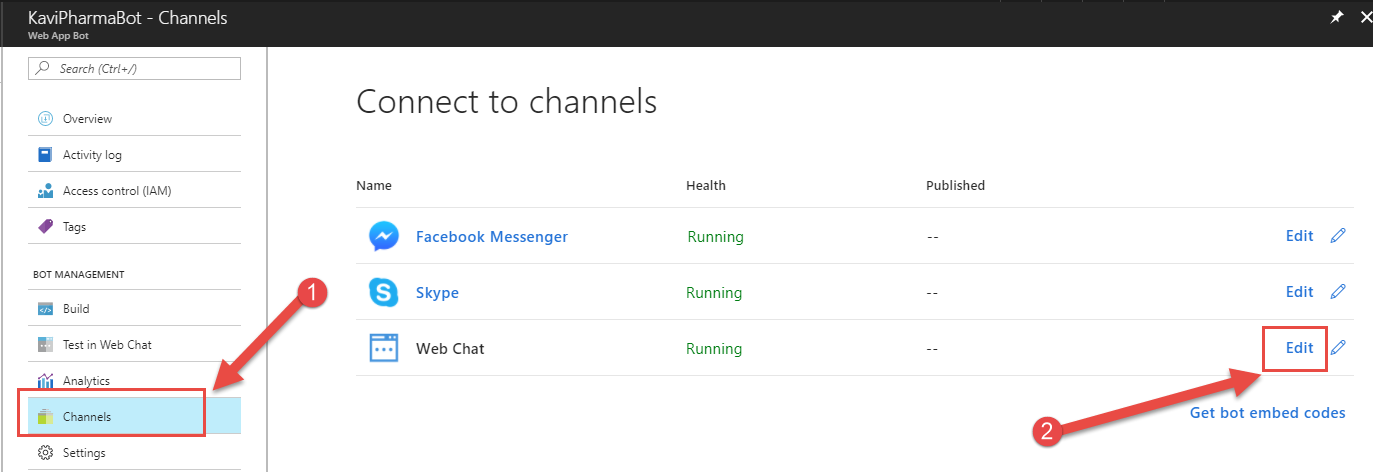


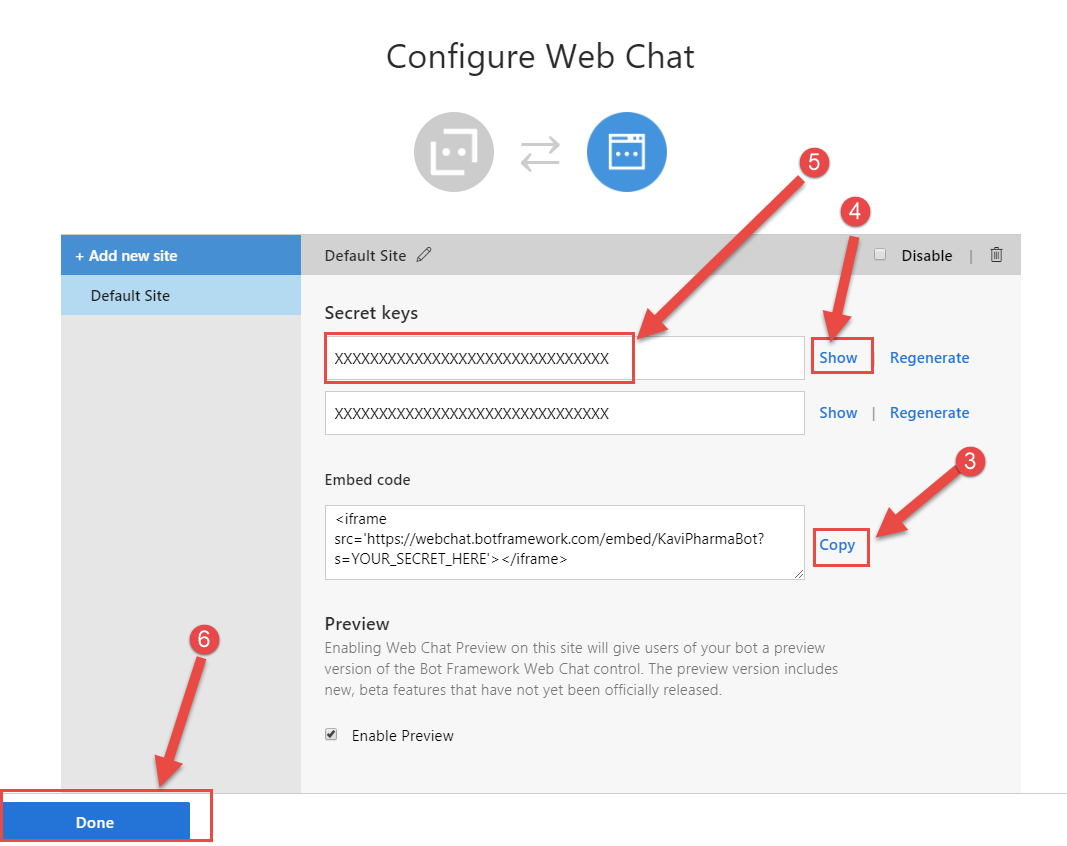
## From Skype, search for KaviPharmaBot and send a few messages to test



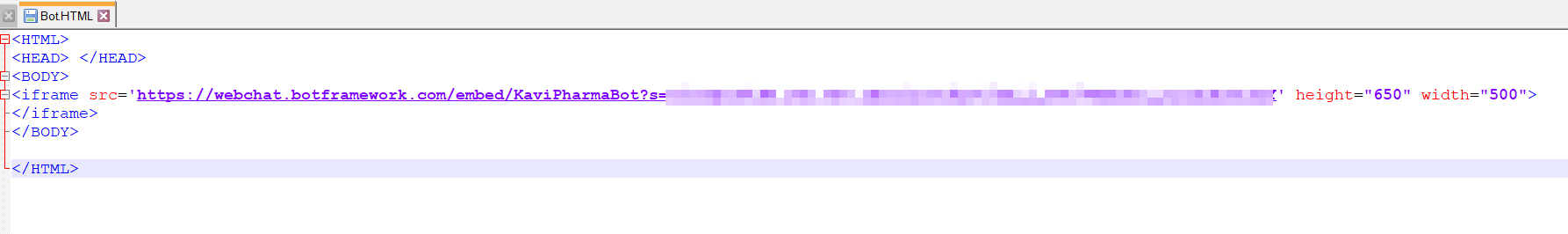
## Connecting to Bot through a Webpage

1. On Azure Portal, follow the steps below

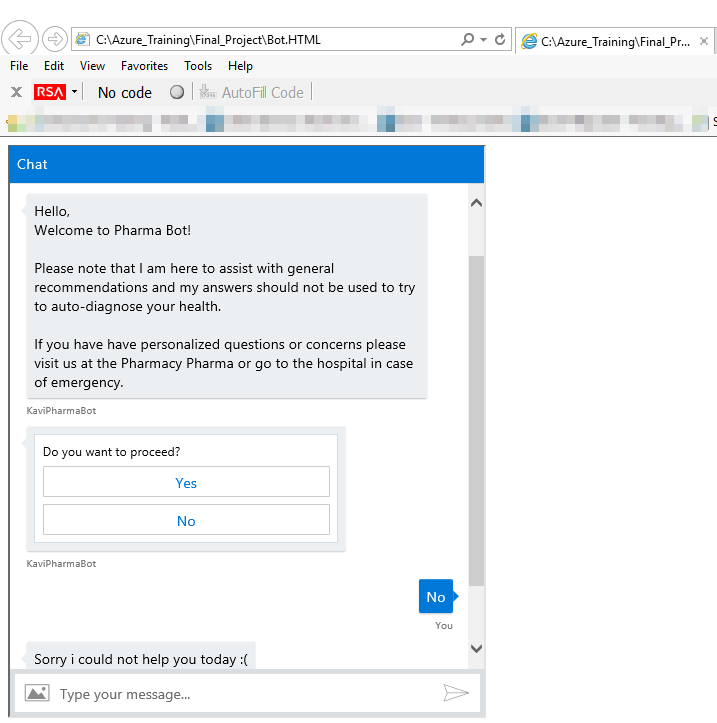




1. Create an HTML page using the information from steps 3 and 5 as shown below



1. Open HTML page and use the chat to communicate with the bot



# Summary

## Lesson Learned

Online code editor is not stable. It would often give errors while re-building the code even if there were no errors in the code. (A common error was related to the npm module on the VM running the code)

## Advantages of Azure Bot Services

The use of the bots is very flexible as it allows uses on multiple platforms to connect to it. This means that the users most of the time will not have to install a separate application only to communicate with the bot.

Allows users to ask questions and get a short answer as opposed to a huge article where the users need to search for the information they are looking for.

Allow developers to connect to multiple other Artificial Intelligence API such as FaceAPI, Speech API and LUIS

## Disadvantages of Azure Bot Services

Users might be resistant to the speaking with Bots in the beginning.

While the Bot services allow connection to different platforms, this requiring monitoring the changes that will occur on the other platform to ensure that, the Bot Service on a specific platform is not affected.

## Next Steps

To benefit from all the aspect of the Azure Bot, we could integrate use Forms to capture user’s Information and proceed to assist the user. Next time the visit the BOT, the user could use the Face API to identify himself. With LUIS we could track the behavior of the user and help determine whether to transfer the user to a human (transferring the user profile and the information previously capture).The speech API could also be used for input of the user as well as receiving the answers from the Bot. If we are able to merge all these functionalities, we will end up with a Bot which can identify a user, assist him and escalate to a real person if needed.

## References

<https://www.webmd.com/cold-and-flu/top-10-questions-cold>

<https://docs.microsoft.com/en-us/bot-framework/bot-service-quickstart>

<https://docs.microsoft.com/en-us/bot-framework/bot-service-channel-connect-facebook>

<https://www.youtube.com/watch?v=1v_VgTy6q0Y>

# YouTube URLs, GitHub URL

## Short video:

<https://youtu.be/H-TqfHbwMVE>

## Long video:

<https://youtu.be/m-7m0mIRagg>

## GitHub Repository

<https://github.com/kavishen/AzureFinal>