Table of Contents

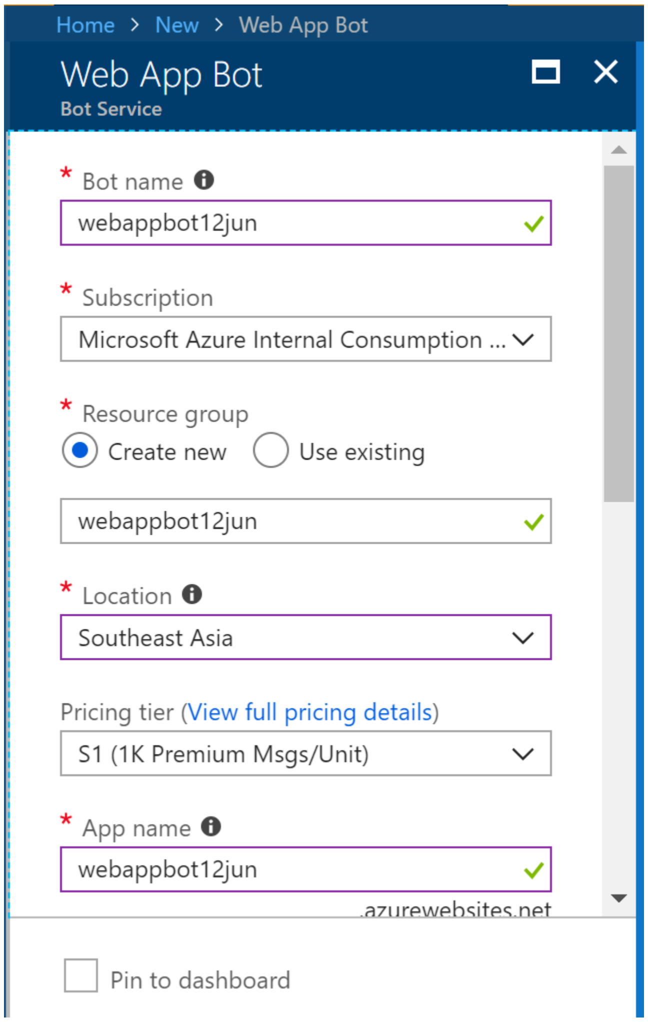
Overview Deck

Microsoft Bot Framework Overview.

Hands on Lab

Create Bot

1. Go to Azure portal
2. Create Web App Bot



Home > New > Web App Bot

Web App Bot
Bot Service

* Bot name: webappbot12jun ✓

* Subscription: Microsoft Azure Internal Consumption ...

* Resource group: [● Create new] ○ Use existing

webappbot12jun ✓

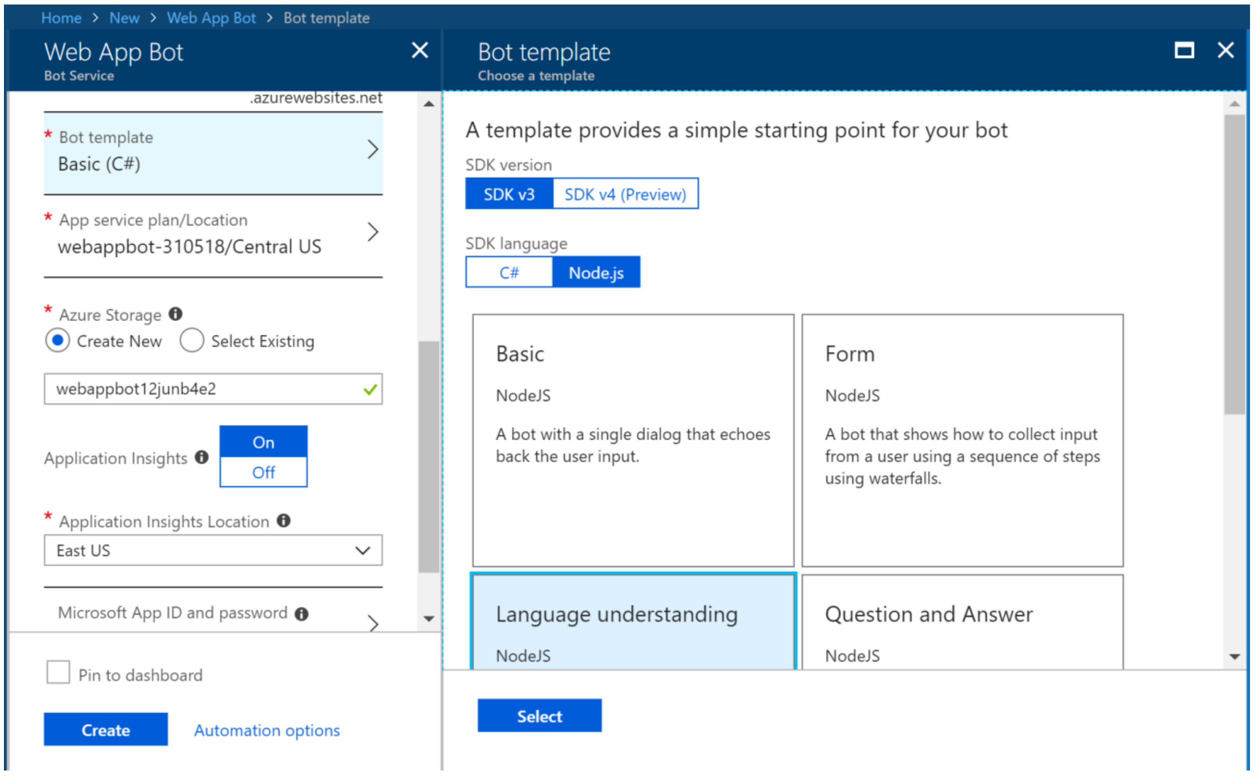
* Location: Southeast Asia

Pricing tier (View full pricing details): S1 (1K Premium Msgs/Unit)

* App name: webappbot12jun ✓
.azurewebsites.net

☐ Pin to dashboard

3. Select NodeJS – LUIS bot as a template and click Create.



.azurewebsites.net

* Bot template
Basic (C#)

* App service plan/Location
webappbot-310518/Central US

* Azure Storage
[✓ Create New] Select Existing

webappbot12junb4e2

Application Insights [✓ On] Off

* Application Insights Location
East US

Microsoft App ID and password

Pin to dashboard

A template provides a simple starting point for your bot

SDK version
[✓ SDK v3] SDK v4 (Preview)

SDK language
C# [✓ Node.js]

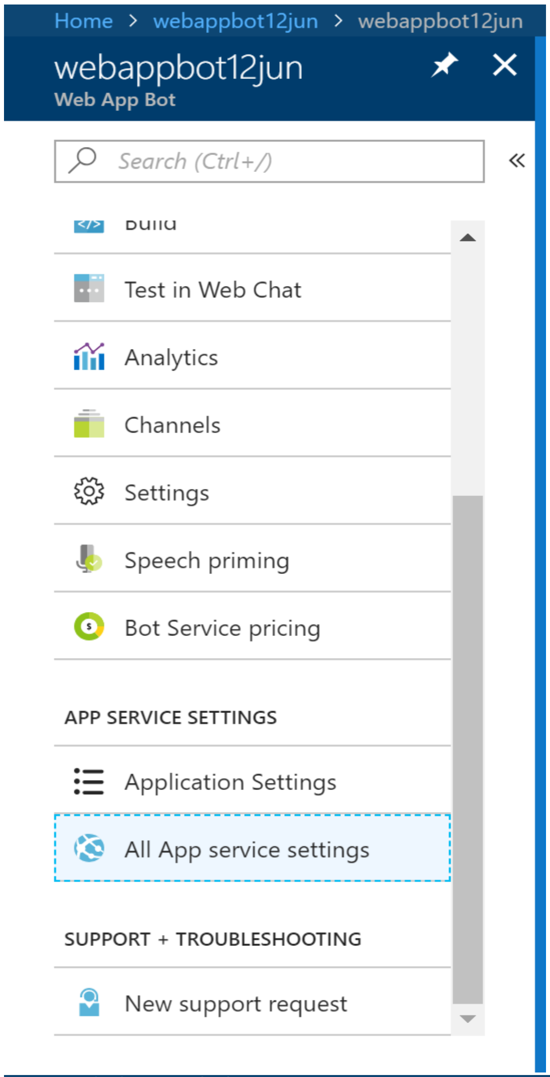
Basic
NodeJS
A bot with a single dialog that echoes back the user input.

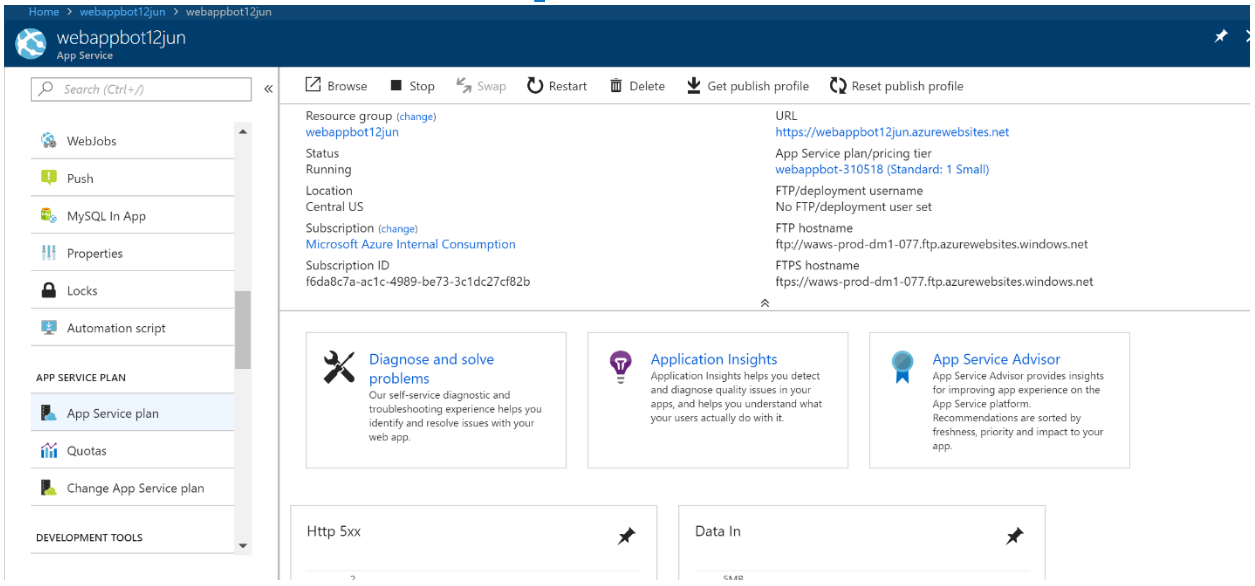
Form
NodeJS
A bot that shows how to collect input from a user using a sequence of steps using waterfalls.

Language understanding
NodeJS

Question and Answer
NodeJS

4. Once the Web app bot is created, navigate to the bot. Navigate to All App Service Settings > App Service Plan > Scale up (App Service Plan). This should show you infrastructure currently used for hosting the Bot.





Resource group (change)
webappbot12jun

Status
Running

Location
Central US

Subscription (change)
Microsoft Azure Internal Consumption

Subscription ID
f6da8c7a-ac1c-4989-be73-3c1dc27cf82b

URL
https://webappbot12jun.azurewebsites.net

App Service plan/pricing tier
webappbot-310518 (Standard: 1 Small)

FTP/deployment username
No FTP/deployment user set

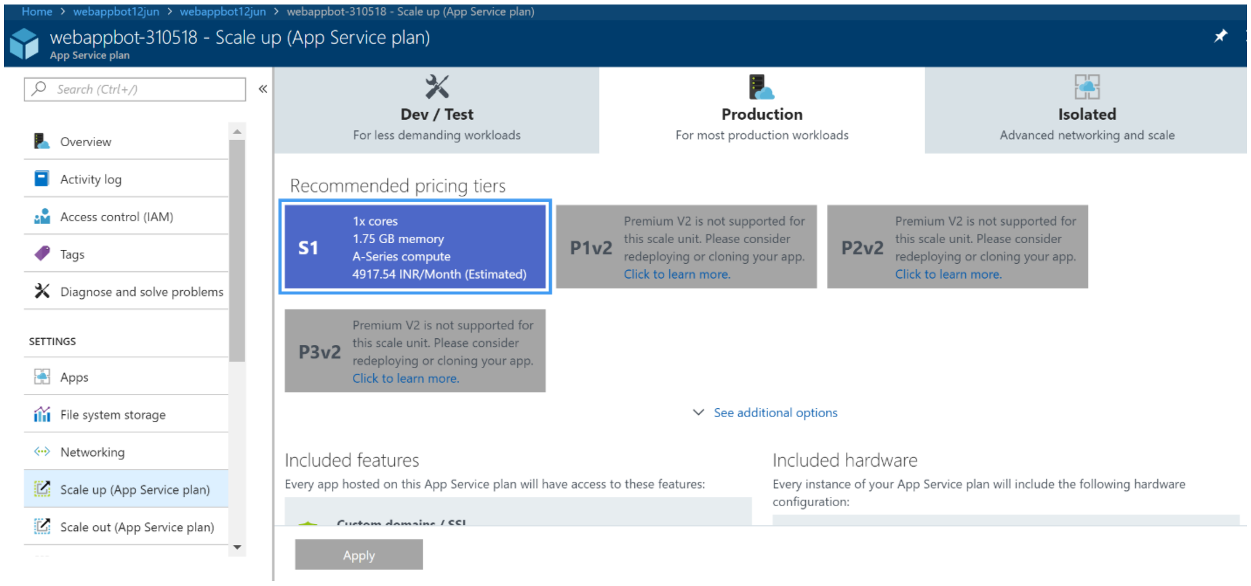
FTP hostname
ftp://waws-prod-dm1-077.ftp.azurewebsites.windows.net

FTPS hostname
ftps://waws-prod-dm1-077.ftp.azurewebsites.windows.net

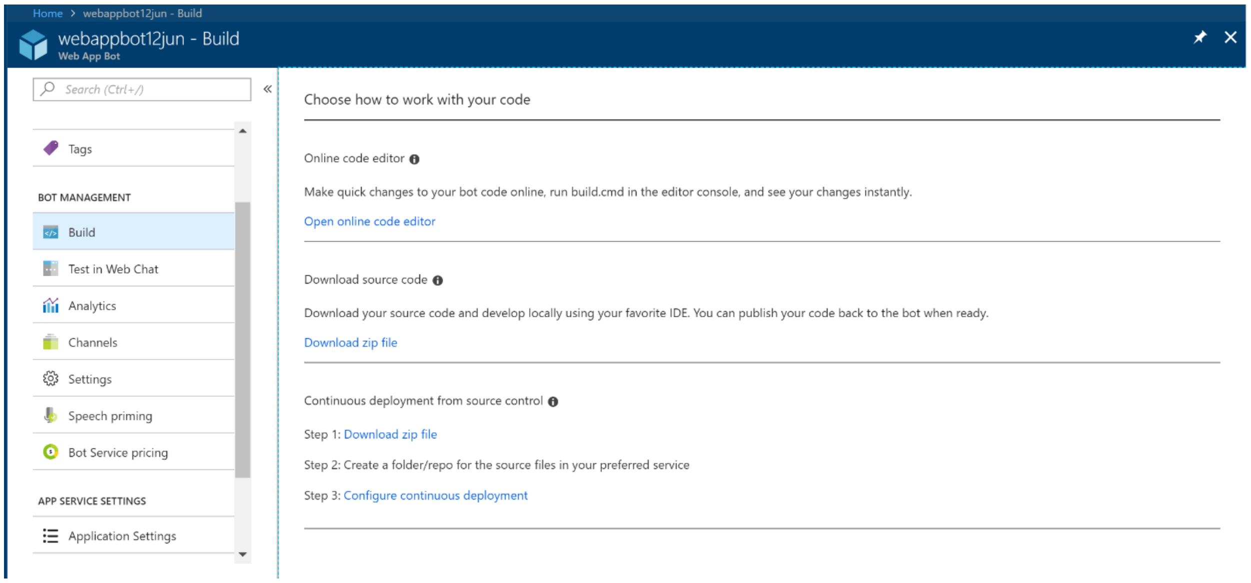
Diagnose and solve problems
Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

Application Insights
Application Insights helps you detect and diagnose quality issues in your apps, and helps you understand what your users actually do with it.

App Service Advisor
App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.


Home > webappbot12jun > webappbot12jun > webappbot-310518 - Scale up (App Service plan)
webappbot-310518 - Scale up (App Service plan)
App Service plan
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
SETTINGS
Apps
File system storage
Networking
Scale up (App Service plan)
Scale out (App Service plan)
Dev / Test
For less demanding workloads
Production
For most production workloads
Isolated
Advanced networking and scale
Recommended pricing tiers
S1
1x cores
1.75 GB memory
A-Series compute
4917.54 INR/Month (Estimated)
P1v2
Premium V2 is not supported for this scale unit. Please consider redeploying or cloning your app. Click to learn more.
P2v2
Premium V2 is not supported for this scale unit. Please consider redeploying or cloning your app. Click to learn more.
P3v2
Premium V2 is not supported for this scale unit. Please consider redeploying or cloning your app. Click to learn more.
See additional options
Included features
Every app hosted on this App Service plan will have access to these features:
Included hardware
Every instance of your App Service plan will include the following hardware configuration:
Apply

5. Navigate back to the Bot and Click on Build menu. Download Zip file



Choose how to work with your code

Online code editor

Make quick changes to your bot code online, run build.cmd in the editor console, and see your changes instantly.

Open online code editor

Download source code

Download your source code and develop locally using your favorite IDE. You can publish your code back to the bot when ready.

Download zip file

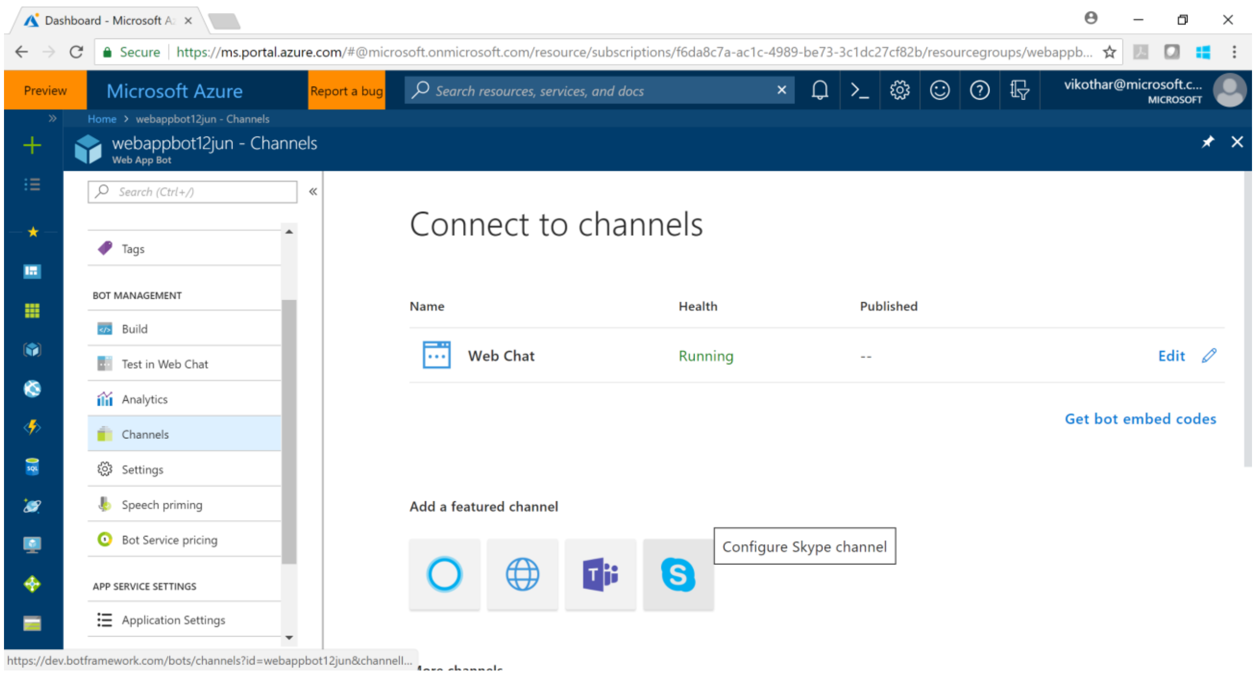
Continuous deployment from source control

Step 1: Download zip file

Step 2: Create a folder/repo for the source files in your preferred service

Step 3: Configure continuous deployment

6. Navigate to Channels, Select Web Chat, Click on Edit


Dashboard - Microsoft A
https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/f6da8c7a-ac1c-4989-be73-3c1dc27cf82b/resourcegroups/webappb...
Preview
Microsoft Azure
Report a bug
Search resources, services, and docs
vikothar@microsoft.c...
MICROSOFT
Home > webappbot12jun - Channels
webappbot12jun - Channels
Web App Bot
Search (Ctrl+/)
Tags
BOT MANAGEMENT
Build
Test in Web Chat
Analytics
Channels
Settings
Speech priming
Bot Service pricing
APP SERVICE SETTINGS
Application Settings
https://dev.botframework.com/bots/channels?id=webappbot12jun&channell...

Connect to channels

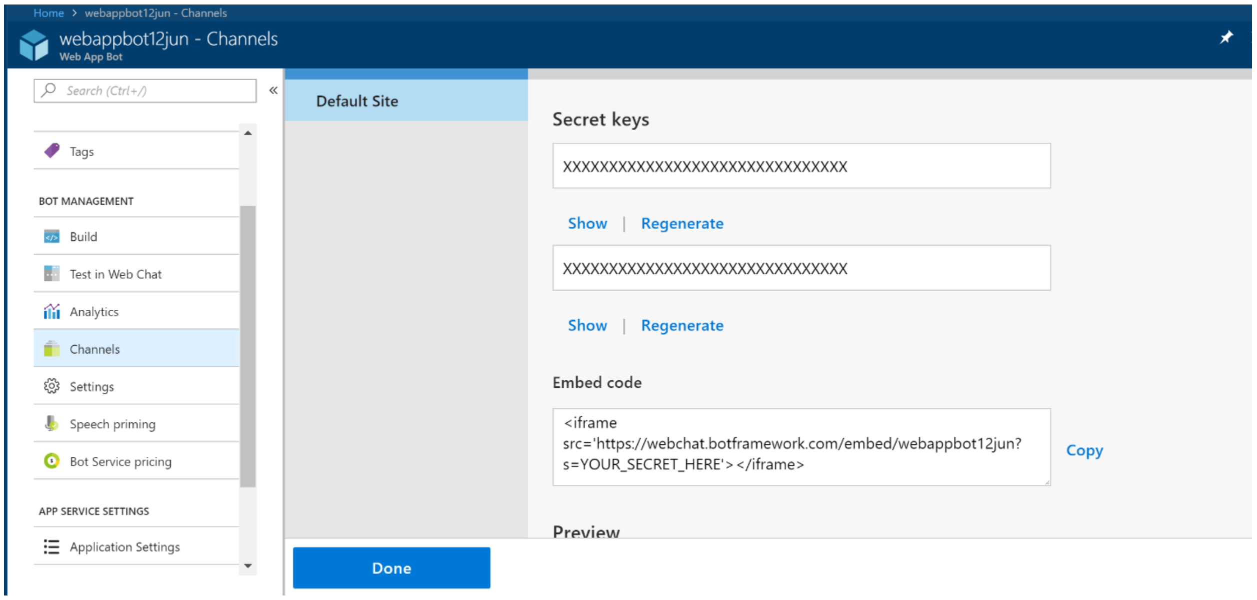
Name	Health	Published	
Web Chat	Running	--	Edit

Get bot embed codes

Add a featured channel

Configure Skype channel

7. Copy Embed code and Secret Key



Secret keys

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

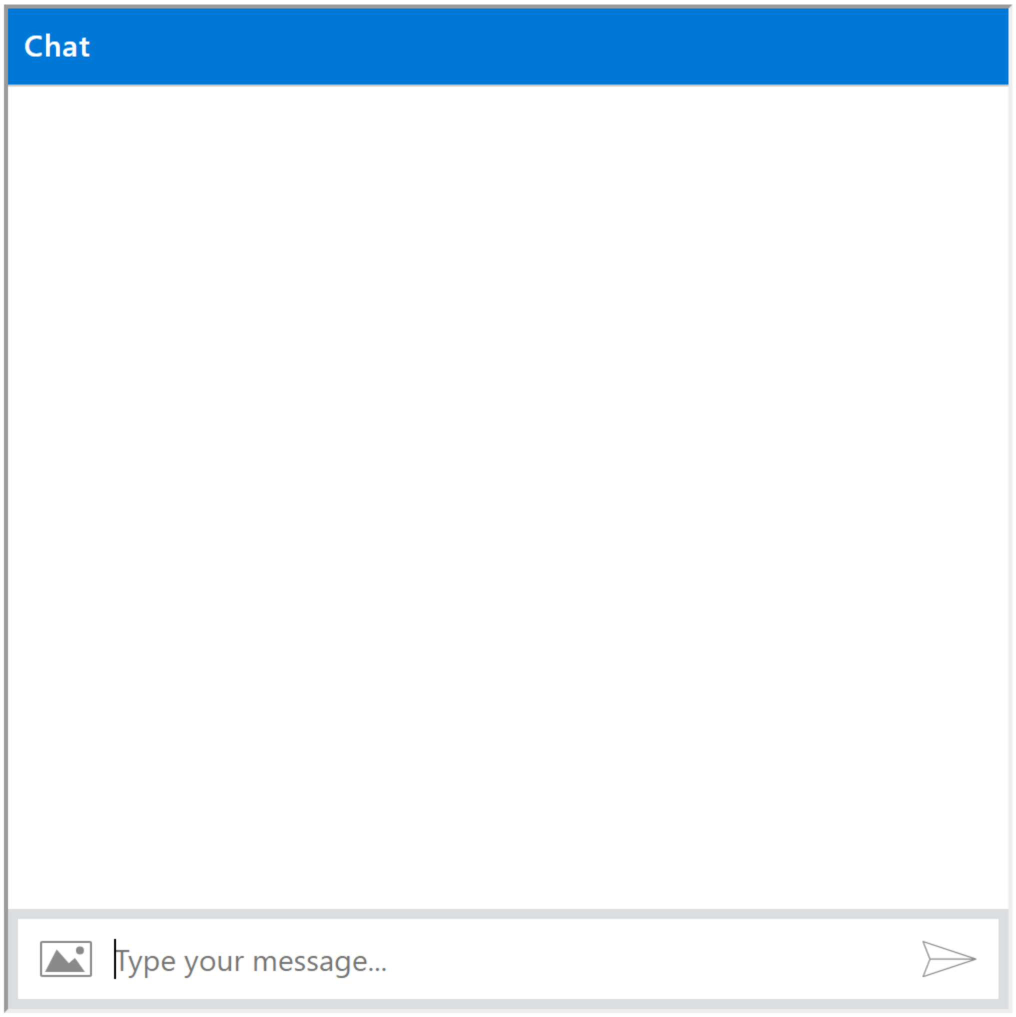
Embed code

```
<iframe
src='https://webchat.botframework.com/embed/webappbot12jun?
s=YOUR_SECRET_HERE'></iframe>
```

8. Create a .html file; paste HTML fragment from previous step with the secret key.
   In .html file, add attributes height=500 and width=500 to IFRAME.

```
<iframe width="500" height="500"
src='https://webchat.botframework.com/embed/webappbot-310518?s=<your-
secret-key>'></iframe>
```

9. Open HTML file and it should load webchat control.



10. Type in Hello and Bot should revert “You reached the Greeting intent. You said ‘Hello’.

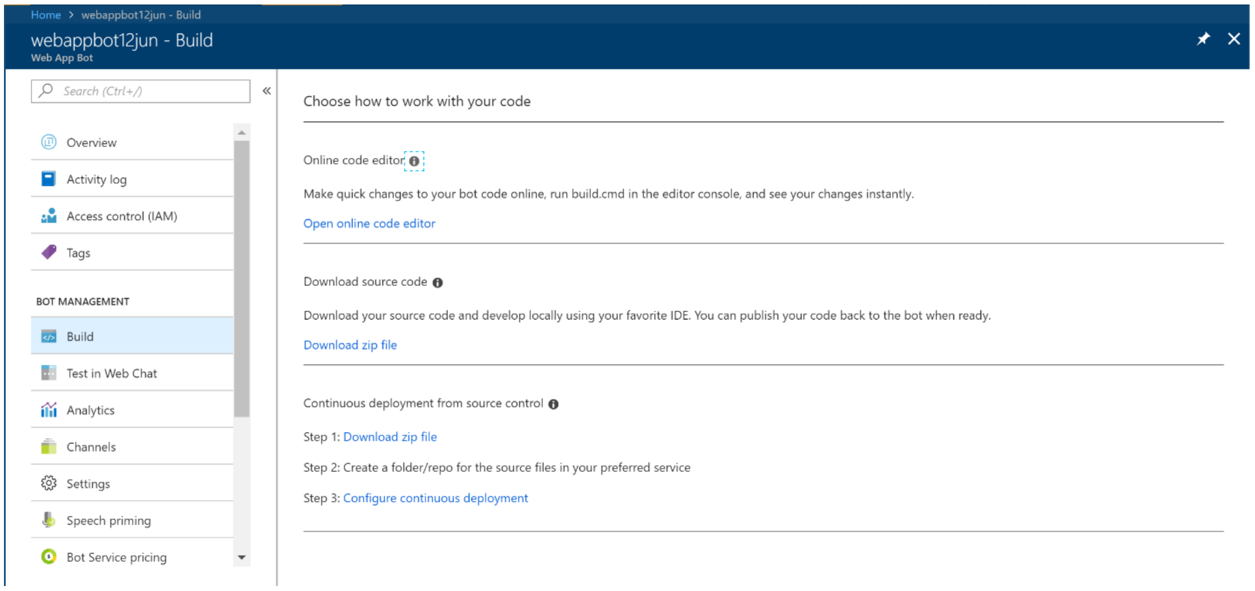
Configure Continuous Integration

11. Create a new repository on Github
12. Add Files to GitHub

```
npm install --save restify
npm install --save botbuilder
npm install --save botbuilder-azure
npm install --save restify
npm install --save botbuilder-cognitiveservices
git init
git add README.md
git add .
git commit -m "first commit"
git remote add origin https://github.com/vishalkothari/azurebotsample-
nodejs.git
```

```
git push -u origin master
```

13. Navigate to Azure portal and to our Web app Bot. Click on Build. Select “Configure continuous deployment”.



Choose how to work with your code

Online code editor

Make quick changes to your bot code online, run build.cmd in the editor console, and see your changes instantly.

Open online code editor

Download source code

Download your source code and develop locally using your favorite IDE. You can publish your code back to the bot when ready.

Download zip file

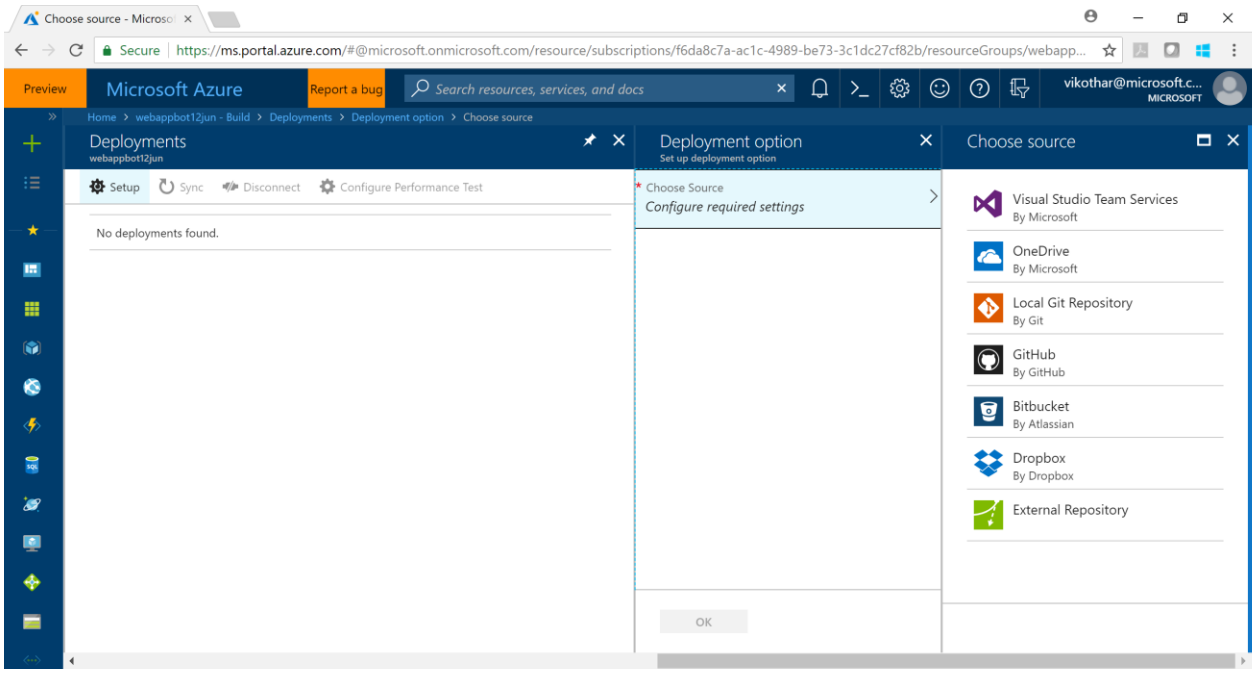
Continuous deployment from source control

Step 1: Download zip file

Step 2: Create a folder/repo for the source files in your preferred service

Step 3: Configure continuous deployment

14. Click on Setup, Choose Source and GitHub



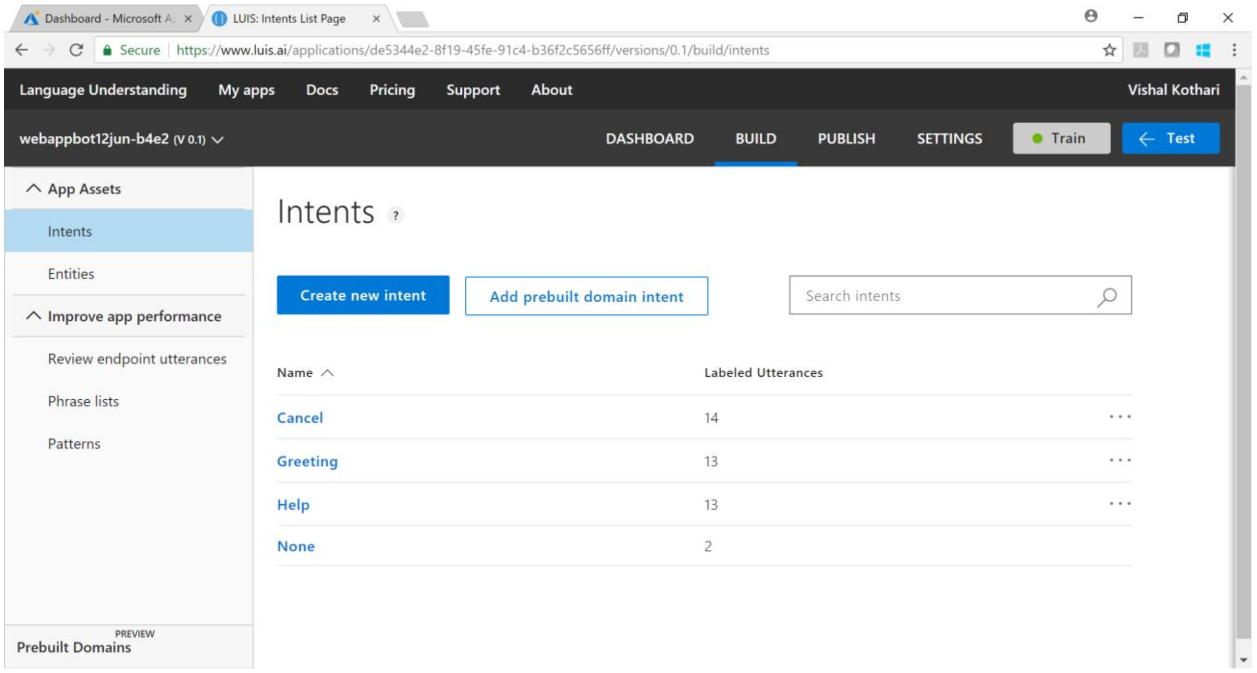
No deployments found.

15. This should prompt for GitHub login and authorization for Azure to access your GitHub repository. If needed, authorization can be revoked from github.com.

Create LUIS Intents

16. Navigate to LUIS portal http://luis.ai and click on “Create LUIS App” or “Go to my apps”

17. Click on the app with same name as the bot created earlier. You should see a screen like below.



Dashboard - Microsoft A | LUIS: Intents List Page

Secure | https://www.luis.ai/applications/de5344e2-8f19-45fe-91c4-b36f2c5656ff/versions/0.1/build/intents

Language Understanding | My apps | Docs | Pricing | Support | About | Vishal Kothari

webappbot12jun-b4e2 (V 0.1) | DASHBOARD | BUILD | PUBLISH | SETTINGS | Train | Test

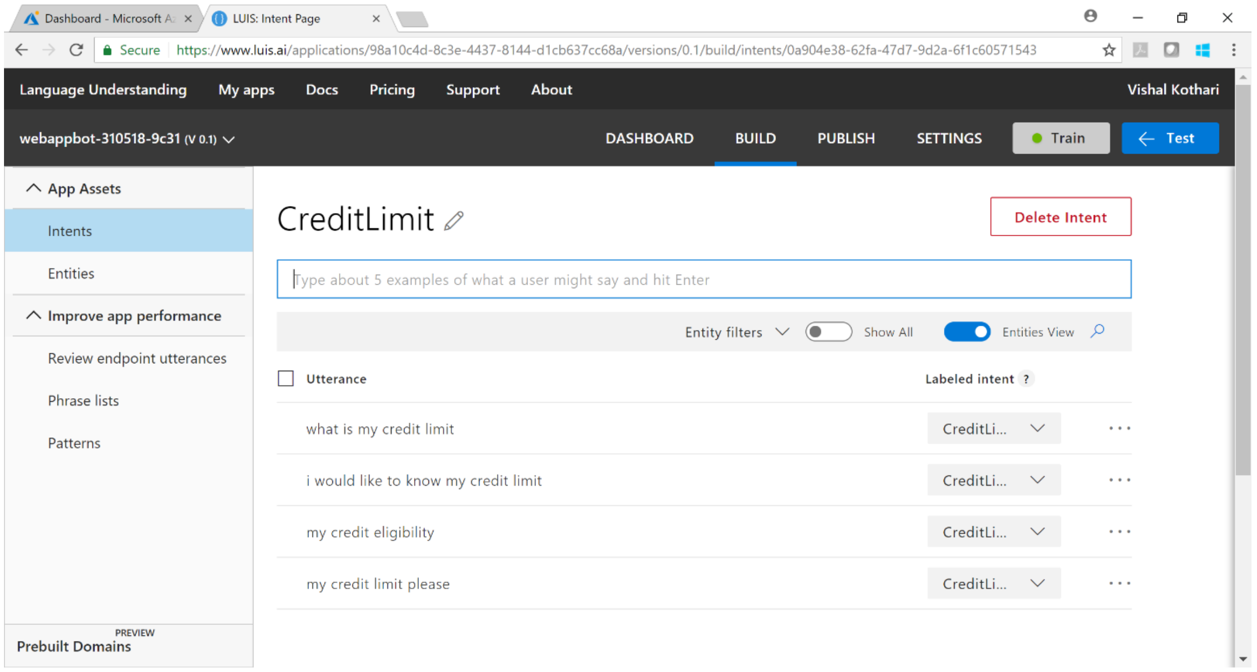
App Assets: Intents, Entities
Improve app performance: Review endpoint utterances, Phrase lists, Patterns
PREVIEW Prebuilt Domains

Intents ?

Create new intent | Add prebuilt domain intent | Search intents

Name	Labeled Utterances
Cancel	14
Greeting	13
Help	13
None	2

18. Click on Create new intent in LUIS.
19. You can add any intent and add some utterances for the intent.



Dashboard - Microsoft A | LUIS: Intent Page

Secure | https://www.luis.ai/applications/98a10c4d-8c3e-4437-8144-d1cb637cc68a/versions/0.1/build/intents/0a904e38-62fa-47d7-9d2a-6f1c60571543

Language Understanding | My apps | Docs | Pricing | Support | About | Vishal Kothari

webappbot-310518-9c31 (V 0.1) | DASHBOARD | BUILD | PUBLISH | SETTINGS | Train | Test

App Assets: Intents, Entities
Improve app performance: Review endpoint utterances, Phrase lists, Patterns
PREVIEW Prebuilt Domains

CreditLimit | Delete Intent

Type about 5 examples of what a user might say and hit Enter

Entity filters | Show All | Entities View

Utterance	Labeled intent ?
what is my credit limit	CreditLi...
i would like to know my credit limit	CreditLi...
my credit eligibility	CreditLi...
my credit limit please	CreditLi...

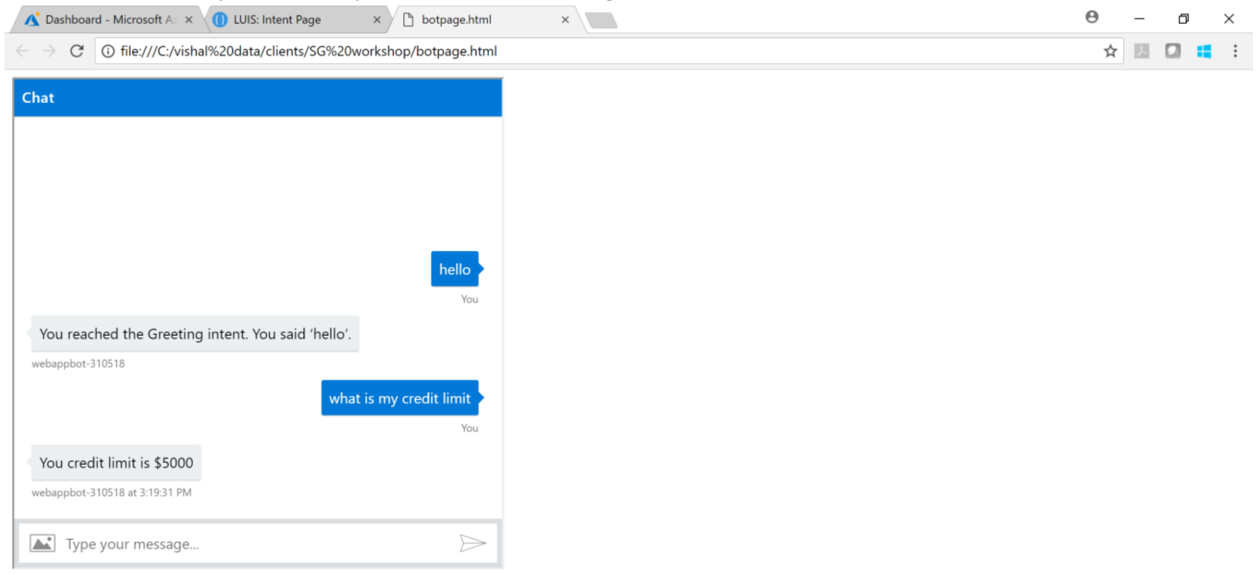
20. Modify the bot code; add a dialog and commit

```
bot.dialog('CreditLimitDialog',
    (session) => {
        session.send('You credit limit is $5000', session.message.text);
        session.endDialog();
    }
).triggerAction({
    matches: 'CreditLimit'
})
```

29. Push code changes to GitHub

```
git add .
git commit -m "added new intent"
git push -u origin master
```

30. The code will be sync'ed and you can see the changes after 2-3 minutes.



hello
You

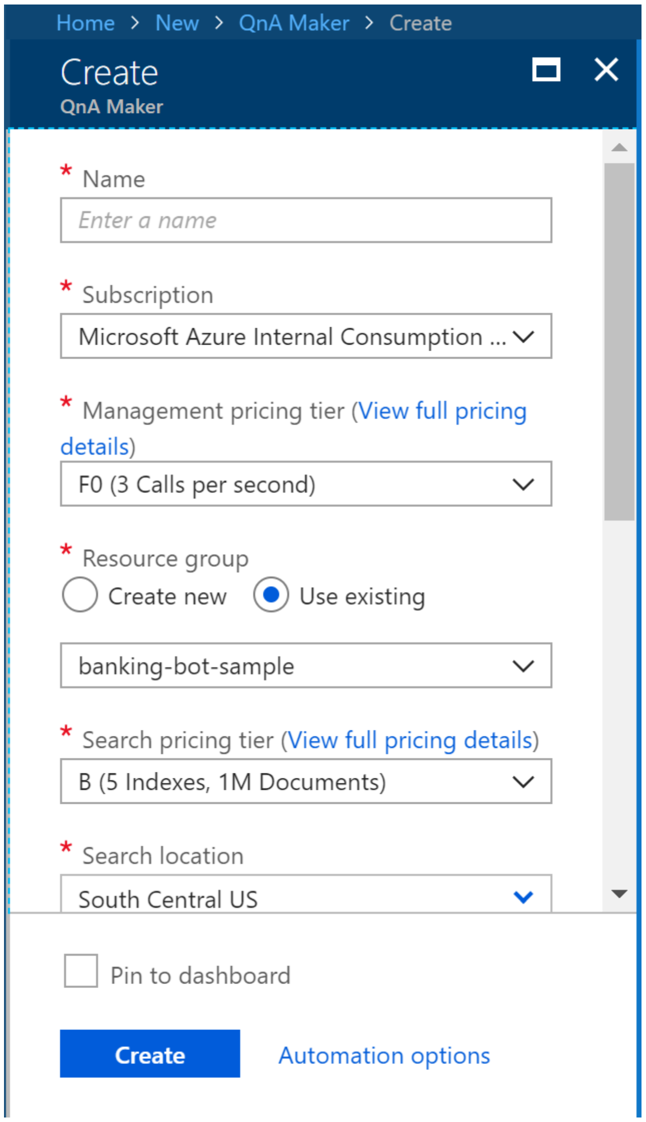
You reached the Greeting intent. You said 'hello'.
webappbot-310518

what is my credit limit
You

You credit limit is $5000
webappbot-310518 at 3:19:31 PM

Create QnA Maker Service

31. Got Azure portal and create new QnA maker service.



Home > New > QnA Maker > Create

Create
QnA Maker

* Name
Enter a name

* Subscription
Microsoft Azure Internal Consumption ...

* Management pricing tier (View full pricing details)
F0 (3 Calls per second)

* Resource group
Create new [✓ Use existing]

banking-bot-sample

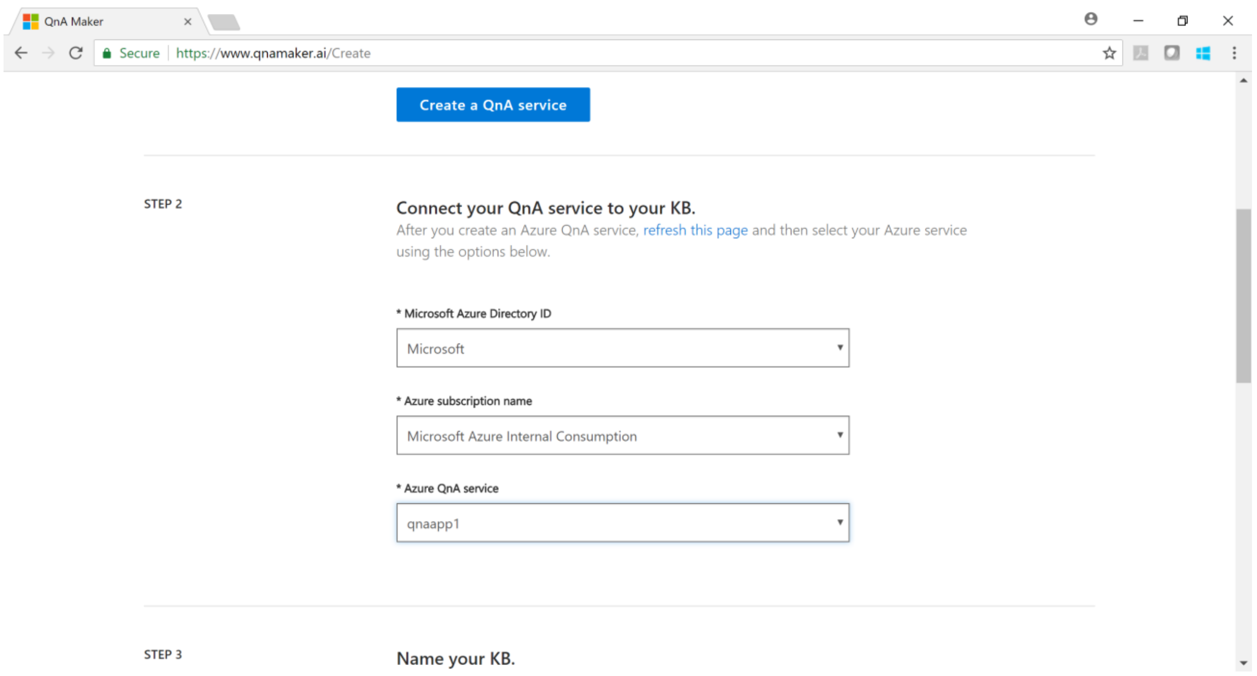
* Search pricing tier (View full pricing details)
B (5 Indexes, 1M Documents)

* Search location
South Central US

Pin to dashboard

Create   Automation options

32. Log on to QnA maker https://www.qnamaker.ai. Click on Create a QnA service.
    Enter FAQ URLs as https://docs.microsoft.com/en-us/azure/virtual-machines/windows/faq-for-disks and https://docs.microsoft.com/en-us/azure/virtual-machines/windows/faq



STEP 2

Connect your QnA service to your KB.
After you create an Azure QnA service, refresh this page and then select your Azure service using the options below.

* Microsoft Azure Directory ID: Microsoft

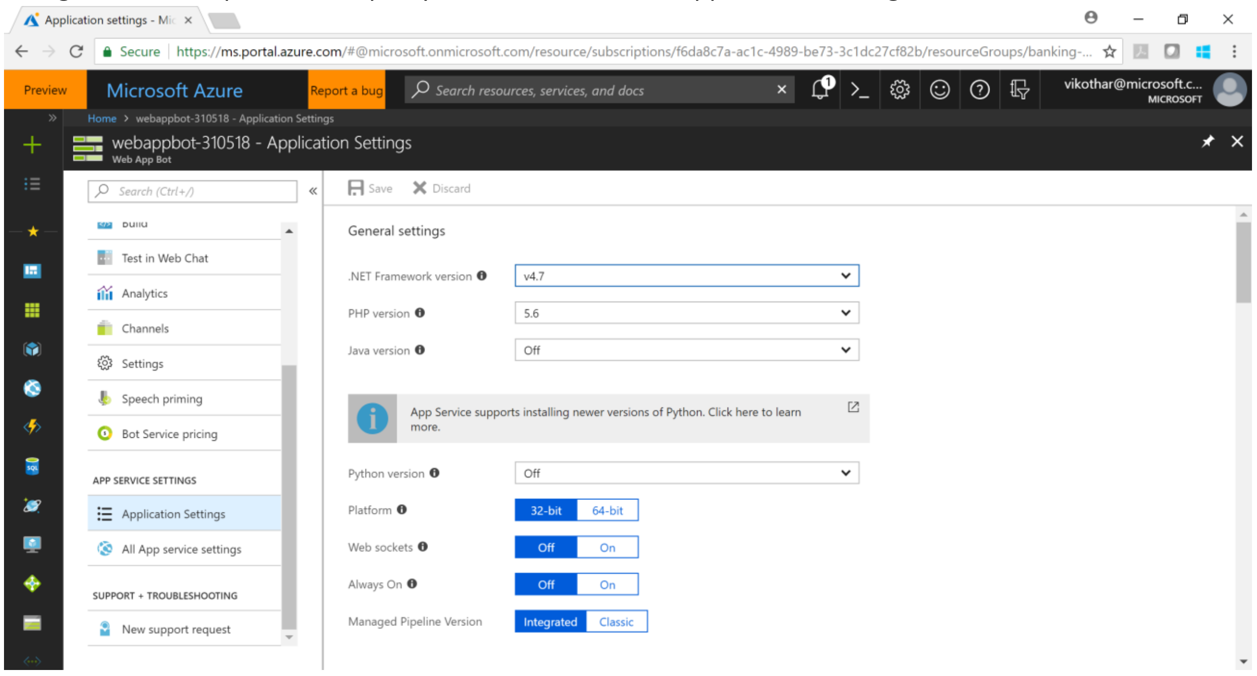
* Azure subscription name: Microsoft Azure Internal Consumption

* Azure QnA service: qnaapp1

STEP 3

Name your KB.

33. Click on Save and Train
34. Go to tab Publish and publish the knowledge base.
35. Navigate to Azure portal and open your Bot and click on Application settings.


Application settings - Mi
Secure | https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/f6da8c7a-ac1c-4989-be73-3c1dc27cf82b/resourceGroups/banking-...
Preview
Microsoft Azure
Report a bug
Search resources, services, and docs
vikothar@microsoft.c...
MICROSOFT
Home > webappbot-310518 - Application Settings
webappbot-310518 - Application Settings
Web App Bot
Search (Ctrl+/)
Build
Test in Web Chat
Analytics
Channels
Settings
Speech priming
Bot Service pricing
APP SERVICE SETTINGS
Application Settings
All App service settings
SUPPORT + TROUBLESHOOTING
New support request
Save
Discard

General settings

.NET Framework version: v4.7

PHP version: 5.6

Java version: Off

App Service supports installing newer versions of Python. Click here to learn more.

Python version: Off

Platform: [✓ 32-bit] 64-bit

Web sockets: [✓ Off] On

Always On: [✓ Off] On

Managed Pipeline Version: [✓ Integrated] Classic

36. In the application settings add variables qnaMakerHost, qnaMakerEndpointKey, qnaMakerKbId and qnaMakerSubscriptionKey and click Save. qnaMakerHost, qnaMakerEndpointKey,

qnaMakerKbId are available in qnamaker.ai and qnaMakerSubscriptionKey is available in Azure portal.

37. Navigate to https://luis.ai and create a new intent called "AzureVMQuestions" and Add utterances Azure VM, Azure Virtual Machine, Linux VM, Linux Virtual Machine, Windows VM, Windows Virtual Machine.
38. Edit to app.js and add

```
var cog = require('botbuilder-cognitiveservices');

var qnaMakerHost = process.env.qnaMakerHost ;
var qnaMakerEndpointKey = process.env.qnaMakerEndpointKey;
var qnaMakerKbId = process.env.qnaMakerKbId;
var qnaMakerSubscriptionKey = process.env.qnaMakerSubscriptionKey;


var qnaRecognizer = new cog.QnAMakerRecognizer({
    knowledgeBaseId: qnaMakerKbId,
    subscriptionKey: qnaMakerSubscriptionKey
});

bot.dialog('AzureVMQuestions', function (session, args) {
    var query = session.message.text;
    cog.QnAMakerRecognizer.recognize(query,
        qnaMakerHost+ '/knowledgebases/' + qnaMakerKbId + '/generateAnswer',
        'EndpointKey ' + qnaMakerEndpointKey, 'Authorization', 1, 'AzureVMQuestions', (error, results) => {
        session.send(results.answers[0].answer);
    })
}).triggerAction({
    matches: 'AzureVMQuestions'
})
```

QnA_knowledgeBaseId and QnA_subscriptionKey are available on Publish tab in QnA maker.

62. Train, Test and publish the LUIS app.
63. Push the code to Git repository and test again.

Create waterfall dialog

64. You can add nested dialogs and richer controls to you bot using Azure bot framework. Open app.js and replace CreditLimitDialog with below code.

```
bot.dialog('CreditLimitDialog', [
    function (session, args, next) {
        session.dialogData.profile = args || {}; // Set the profile or create the object.
        if (!session.dialogData.profile.accountType) {
            builder.Prompts.choice(session, "What's your account type?",
"silver|gold|platinum", { listStyle: 3 });
        } else {
            next(); // Skip if we already have this info.
        }
    },
    function (session, results, next) {
        if (results.response) {
            // Save account type if we asked for it.
            console.log(results.response);
            session.dialogData.profile.accountType = results.response.entity;
        }
        if (!session.dialogData.profile.location) {
            builder.Prompts.text(session, "What is your location?");
        } else {
            next(); // Skip if we already have this info.
        }
    },
    function (session, results) {
        console.log("in next");
        if (results.response) {
            // Save location if we asked for it.
            session.dialogData.profile.location = results.response;
        }
        //console.log(session.dialogData.profile);
        session.send(`Hello credit limit for
${session.dialogData.profile.accountType} and in
${session.dialogData.profile.location} is $10000`);
    }
]).triggerAction({
   matches: 'CreditLimit'
});
```

65. Push the code to GitHub and deploy.

Configure Speech (works in Chrome and Edge browsers)

66. Open Azure portal, Navigate to the Bot that you created in Azure and select Channels. Add a directline channel and copy the secret.

67. In Azure portal, create a new Bing speech service and copy the secret.

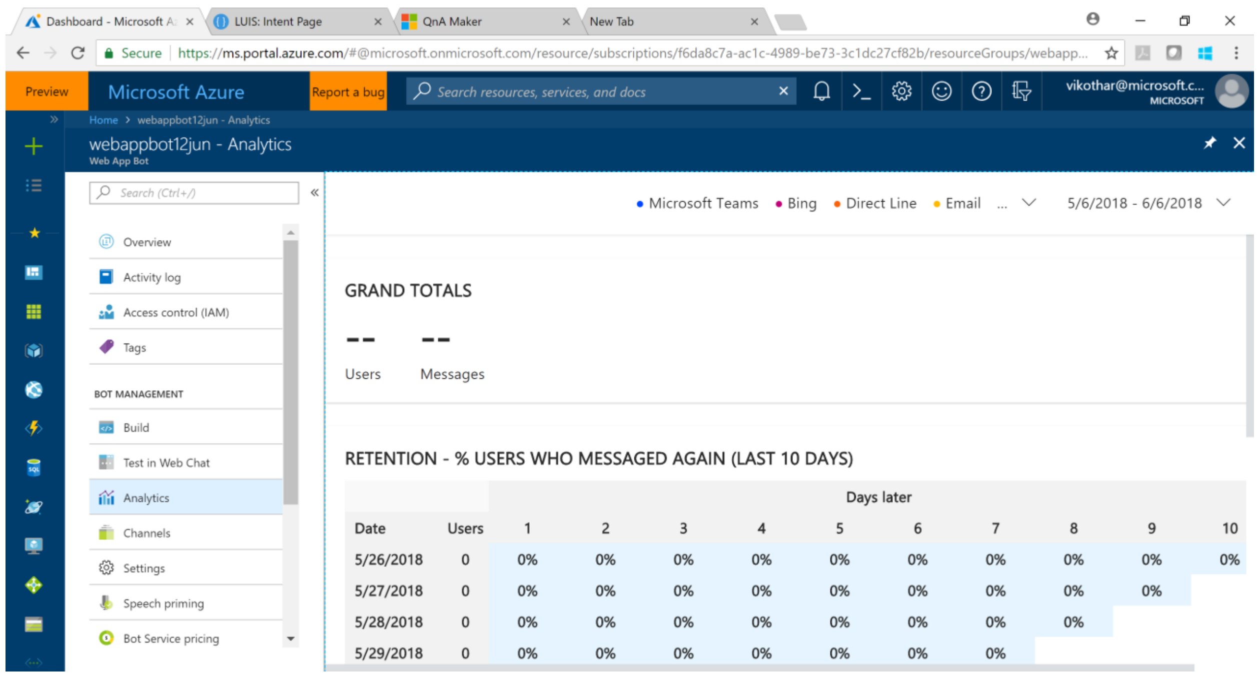
68. Create a new HTML File and add following code and replace your key accordingly.

```
<!DOCTYPE html>
<html>
```

```
  <head>
    <link href="https://cdn.botframework.com/botframework-
webchat/latest/botchat.css" rel="stylesheet" />
  </head>
  <body>
    <div id="bot"/>
    <script src="https://cdn.botframework.com/botframework-
webchat/latest/botchat.js"></script>
    <script src="https://cdn.botframework.com/botframework-
webchat/latest/CognitiveServices.js"></script>

    <script>
      var speechOptions = {
        speechRecognizer: new CognitiveServices.SpeechRecognizer( {
subscriptionKey: '<bing speech key>' } ),
        speechSynthesizer: new CognitiveServices.SpeechSynthesizer(
          {
              subscriptionKey: '<bing speech key>',
              gender: CognitiveServices.SynthesisGender.Female,
              voiceName: 'Microsoft Server Speech Text to Speech Voice (en-
US, JessaRUS)'
          })
      };
      BotChat.App({
        directLine: { secret:
'1dUtH_PtkEg.cwA.HWQ.U8Mlepp0g2dpqsh2QcqvvhZf10LyetnJE0BRfEdDVns' },
        user: { id: 'userid' },
        bot: { id: 'botid' },
        speechOptions: speechOptions,
        resize: 'detect'
      }, document.getElementById("bot"));
    </script>
  </body>
</html>
```

69. Navigate to Web app bot in Azure portal and click on Analytics.
    This gives you a view of number of users, user retention, number messages and channels used in the Bot.



GRAND TOTALS

--	--
Users	Messages

RETENTION - % USERS WHO MESSAGED AGAIN (LAST 10 DAYS)

		Days later									
Date	Users	1	2	3	4	5	6	7	8	9	10
5/26/2018	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5/27/2018	0	0%	0%	0%	0%	0%	0%	0%	0%	0%	
5/28/2018	0	0%	0%	0%	0%	0%	0%	0%	0%		
5/29/2018	0	0%	0%	0%	0%	0%	0%	0%			

Bot navigation design patterns and anti-patterns

Bot navigation antipatterns.pptx

Appendix: To build and debug the bot locally

1. Download and Install Bot Framework Emulator from https://github.com/Microsoft/BotFramework-Emulator/releases

2. Comment out Table storage lines

```
var tableName = 'botdata';
var azureTableClient = new botbuilder_azure.AzureTableClient(tableName,
process.env['AzureWebJobsStorage']);
var tableStorage = new botbuilder_azure.AzureBotStorage({ gzipData: false },
azureTableClient);

bot.set('storage', tableStorage);
```

3. Set the below environment variables.

```
var luisAppId = process.env.LuisAppId ;
var luisAPIKey = process.env.LuisAPIKey;
var qnaMakerHost = process.env.qnaMakerHost ;
var qnaMakerEndpointKey = process.env.qnaMakerEndpointKey;
var qnaMakerKbId =  process.env.qnaMakerKbId;
var qnaMakerSubscriptionKey = process.env.qnaMakerSubscriptionKey;
```

4. Run the bot.

```
node app.js
```

5. Open Bot Framework Emulator. Go to http://localhost:3978/api/messages and click connect.