|  |
| --- |
| \\it.local\data\Teams\konseptointi_tyokansio\Logot\png\digia_logo_cmyk.png |
| chatbot manual  DigisFääri |
|  |

# introduction

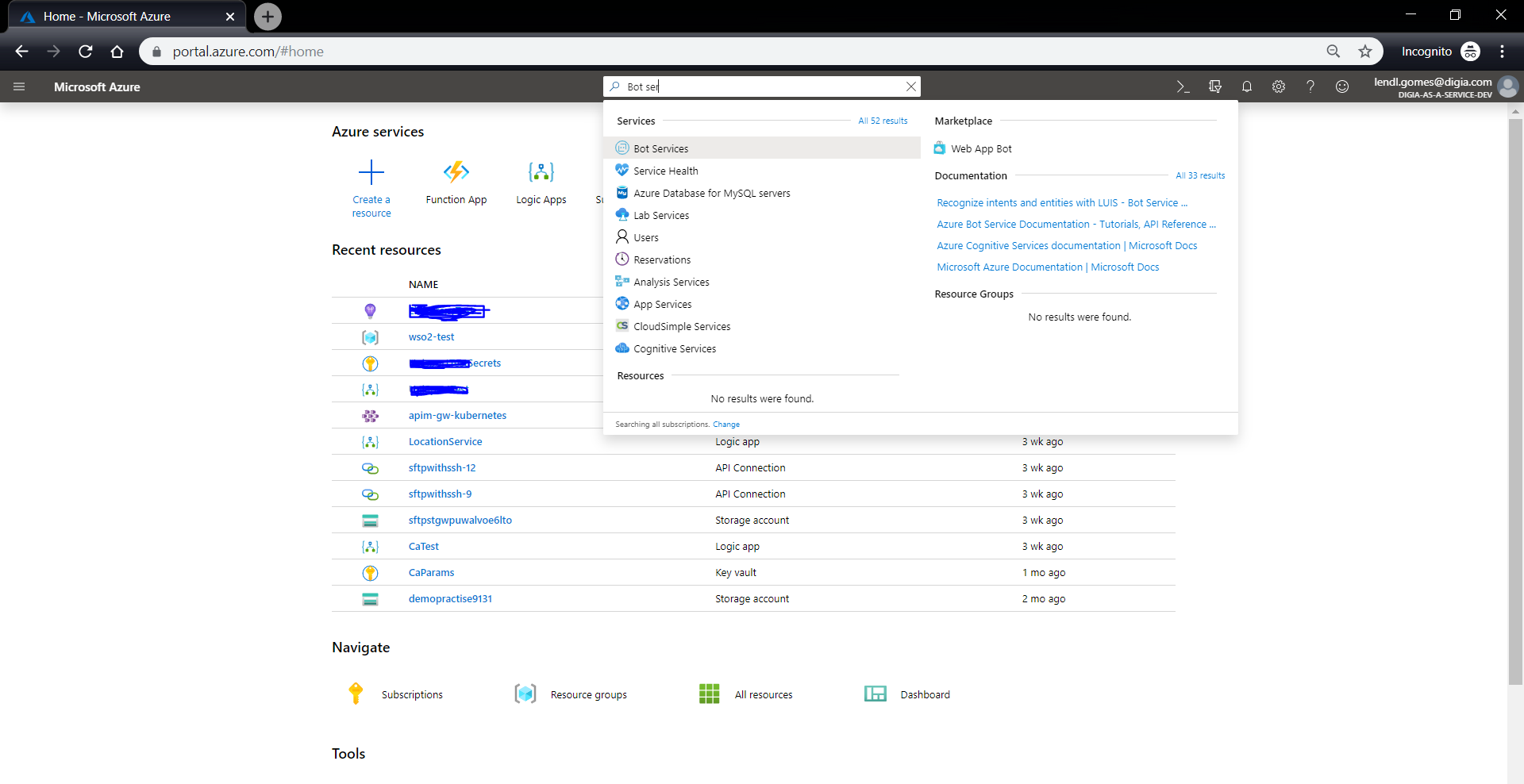
A chatbot is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone.

Azure Bot Service provides an integrated environment that is purpose-built for bot development, enabling you to build, connect, test, deploy, and manage intelligent bots, all from one place.

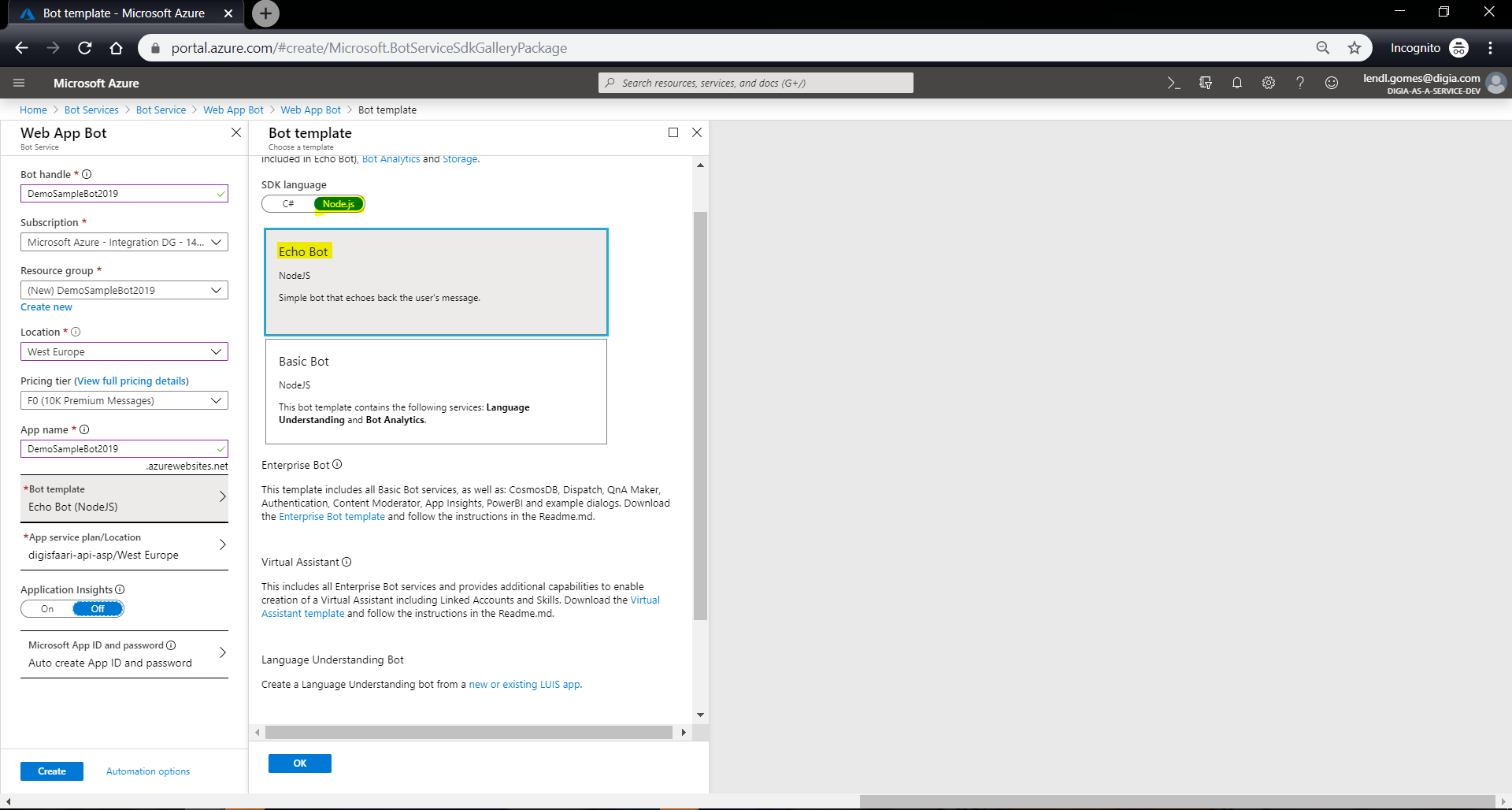
# WorKSHOp

## Build quickstart chat bot from azure portal:

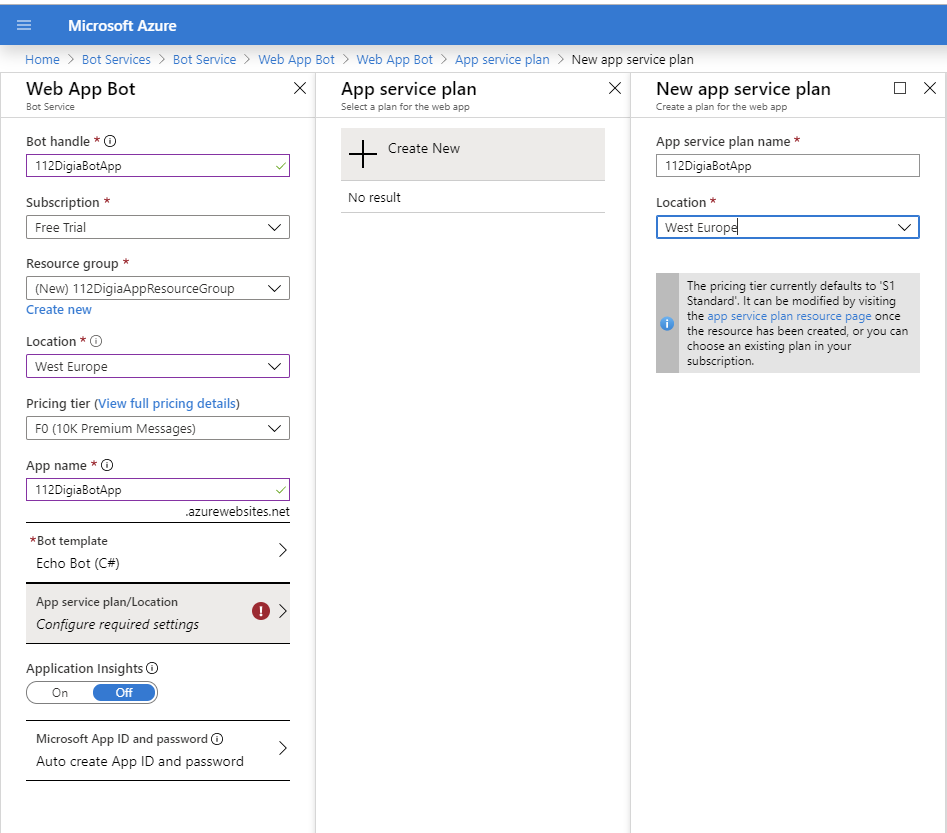
1. Sign in to “[Azure portal](https://portal.azure.com/)” with your azure account and then select “All Services” from the navigation menu.
2. Search for “Bot Services” and go to the services.



1. Click “Add” and select “Web App Bot” and click on “Create”.
2. Fill on the given parameters
3. Bot Handle: Unique name to your bot
4. Subscription: Your subscription
5. Resource group: Select create new and give “your resource group name”
6. Location: West Europe
7. Pricing Tier: Choose F0 since it’s free.
8. Bot Template: **Select Node.js and Echo bot**



1. App service plan/Location: Create new or choose existing one



1. Application Insights: Off
2. Microsoft App ID and password: Auto Create App Id and password
3. Click create

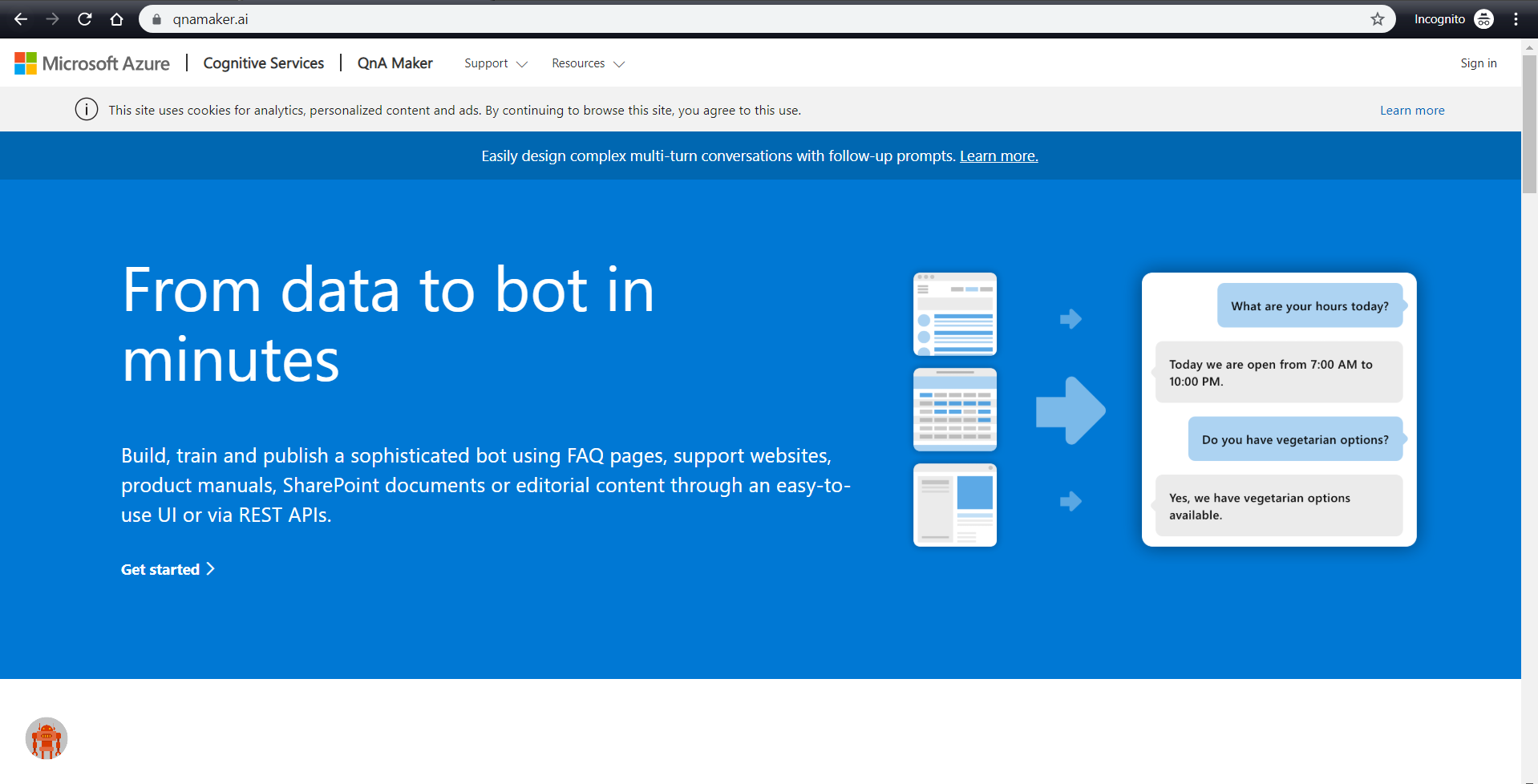
You might end up getting error as “resource provider 'microsoft.web' not registered for the subscription” wait for some time till all required resource gets registered.

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-register-provider-errors>

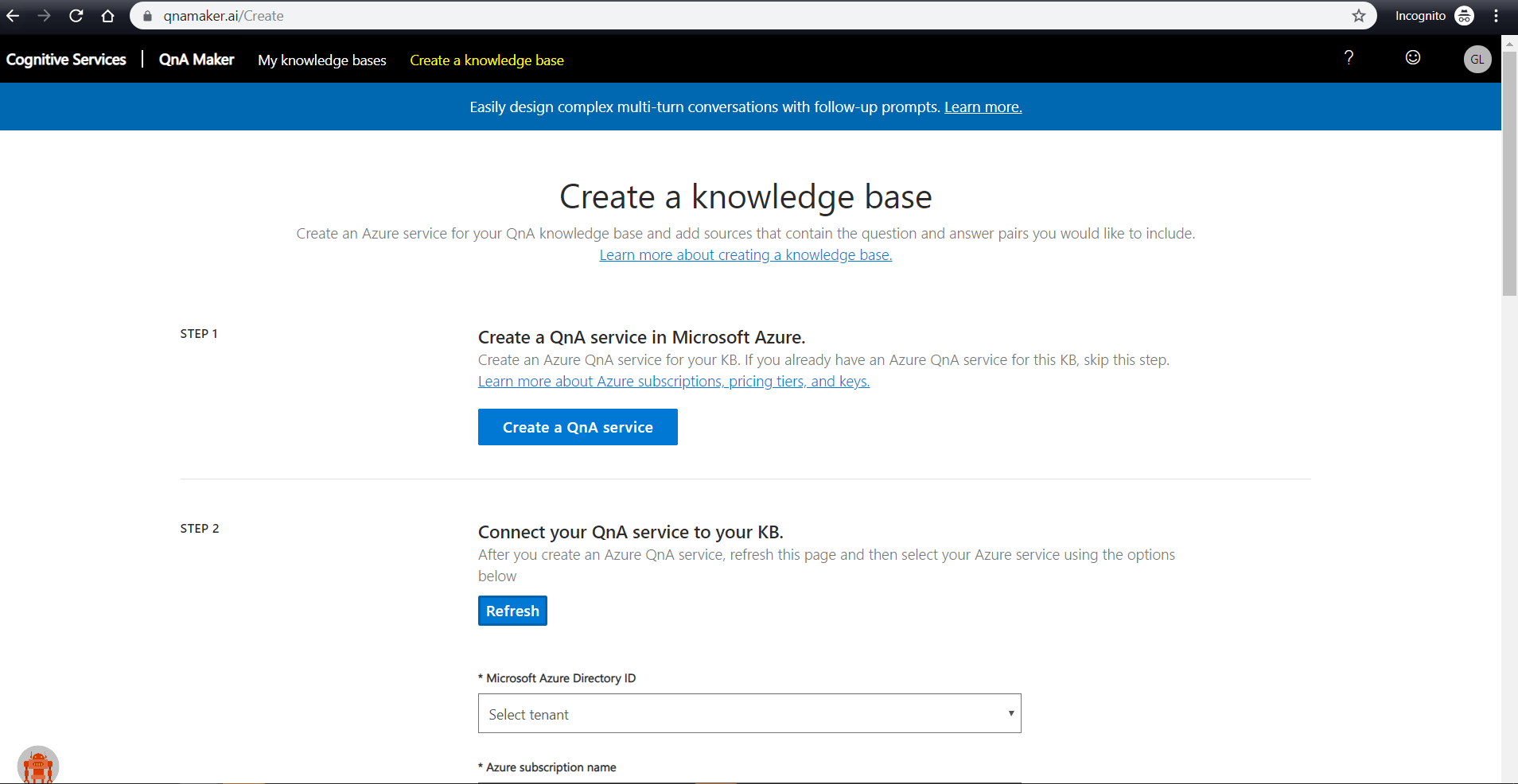
After successful deployment go to Test in Web Chat and type Hi. Chat bot will reply as You said ‘Hi’

## Build QnA knowledge base for digia 112 app:

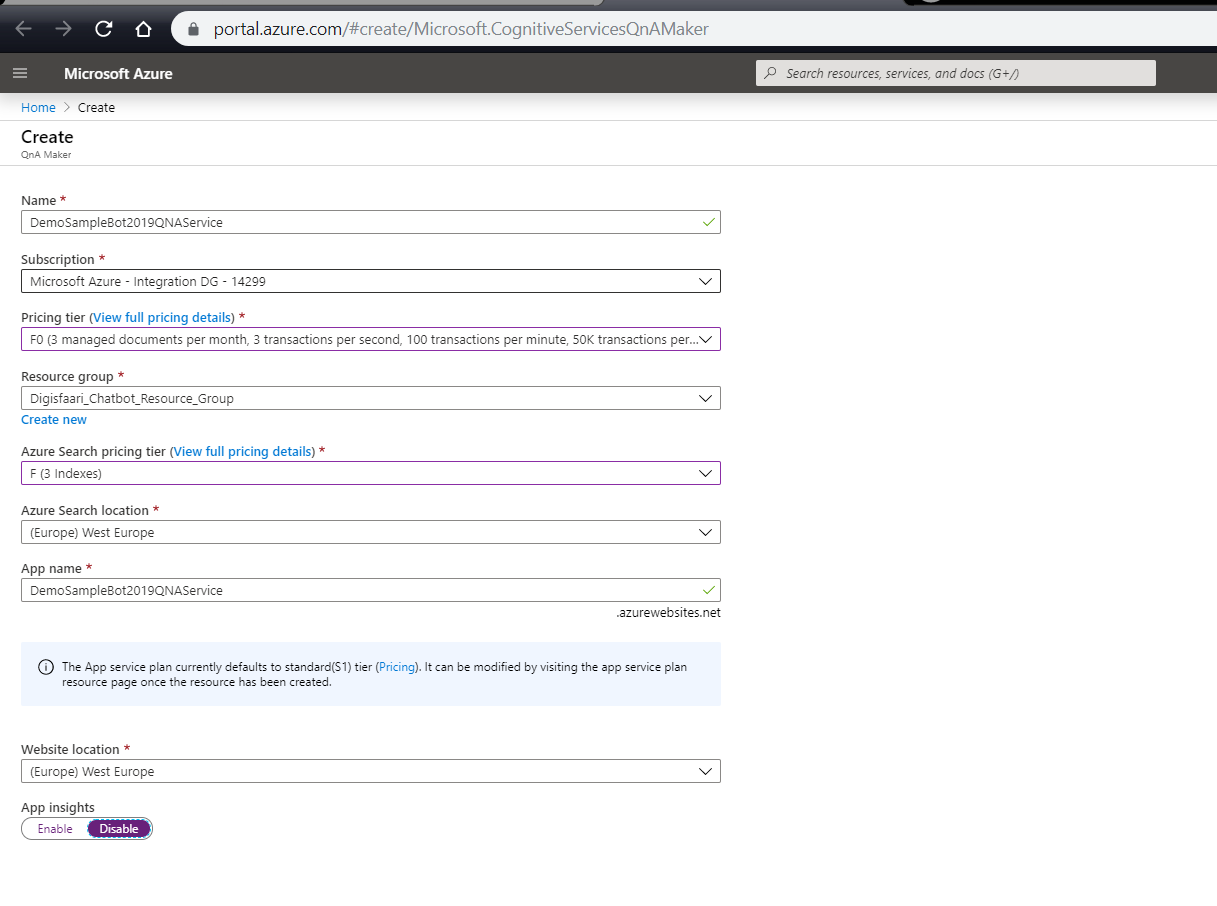
1. Sign into “[QnA maker portal](https://www.qnamaker.ai/)” with your Azure Credentials.



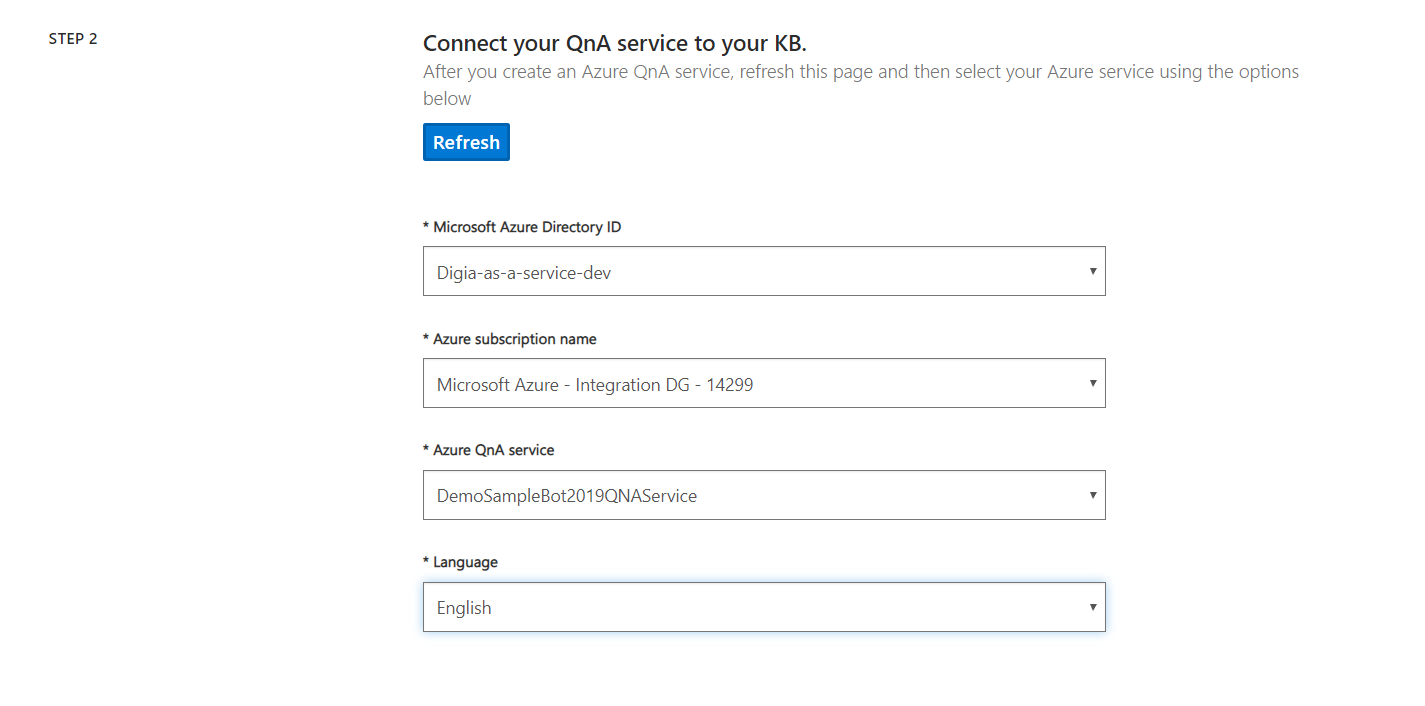
1. In the QnA Maker portal, create a knowledge base.



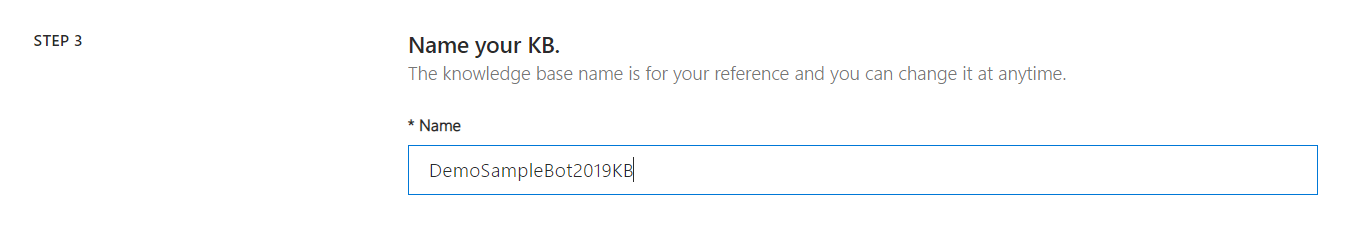
1. Create a QnA service.
2. You will be redirected to your Azure portal then you can create QnA maker Service and fill on given parameters
3. Name: Unique name to your bot
4. Subscription: Your subscription
5. Pricing Tier: F0
6. Resource Group: Your resource group created in previous steps
7. Azure search pricing tier: F
8. Azure search location: West Europe
9. App Name: Type app name here.
10. Website location: West Europe
11. App Insights: disable



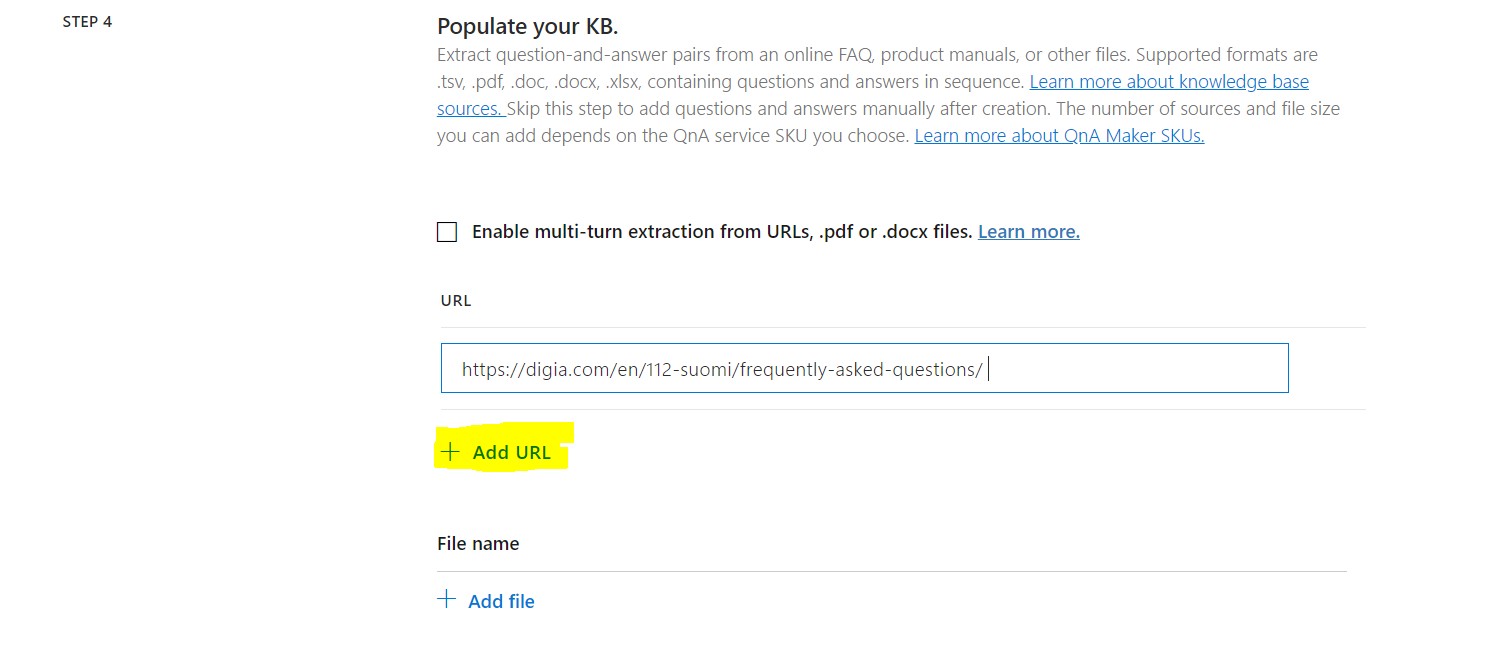
1. Follow the instructions on your QnA maker create page.
2. Connect your QnA service to your knowledge base. (click refresh to see new service created)



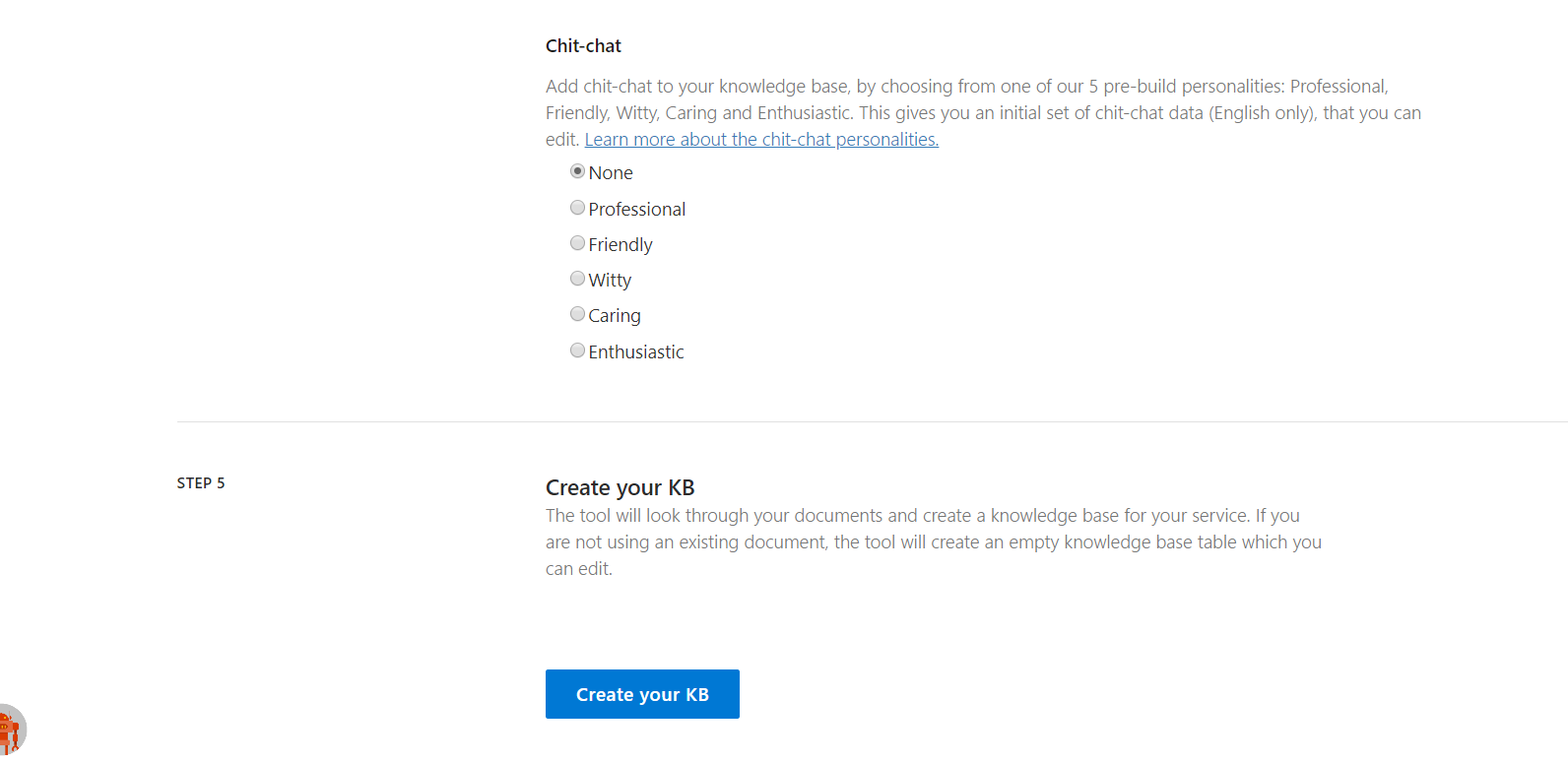
1. Name your knowledge base.



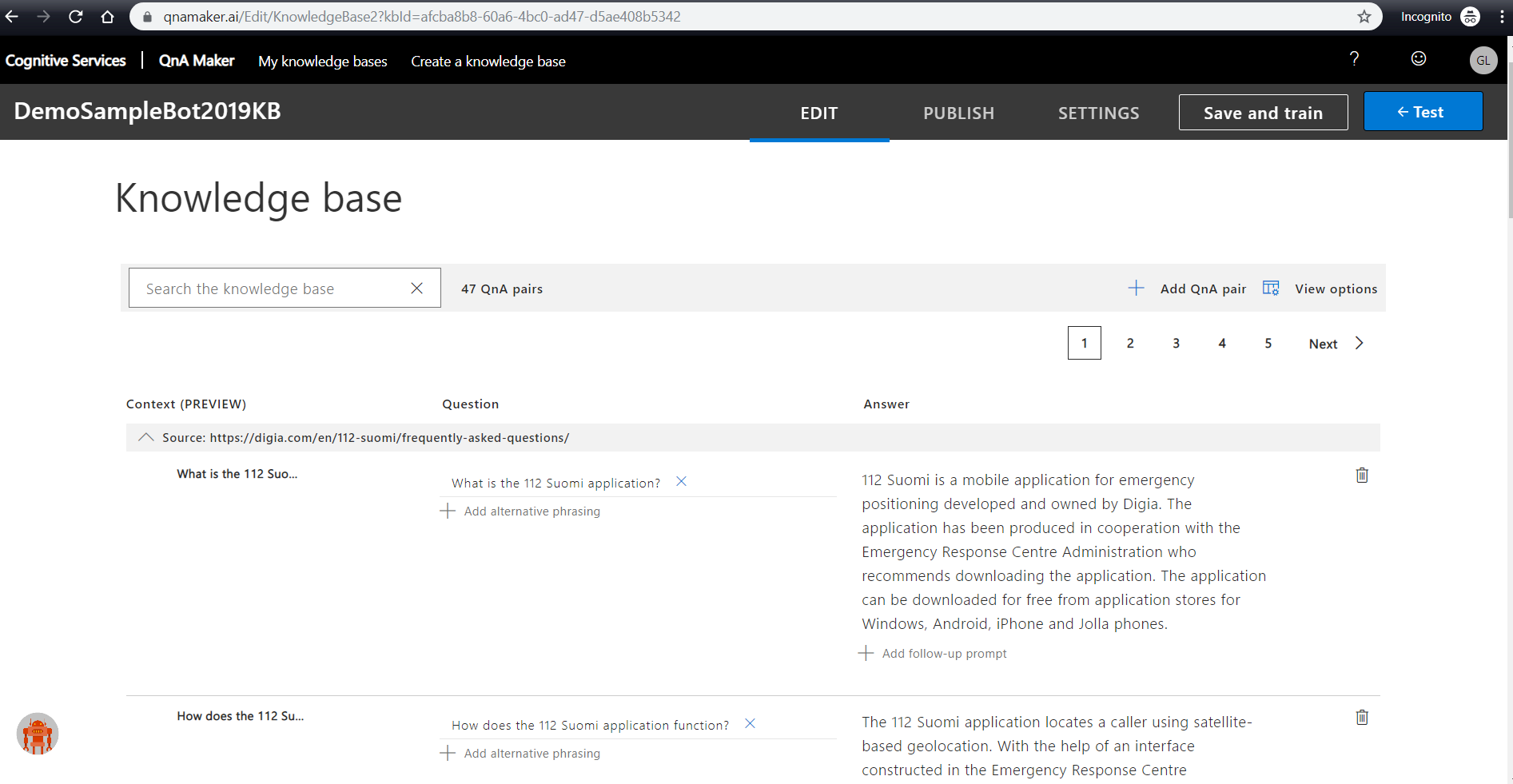
1. To populate your knowledge base, use url (Faqs) or sample given by instructor. (**<https://digia.com/en/112-suomi/frequently-asked-questions/>)** and click Add URL



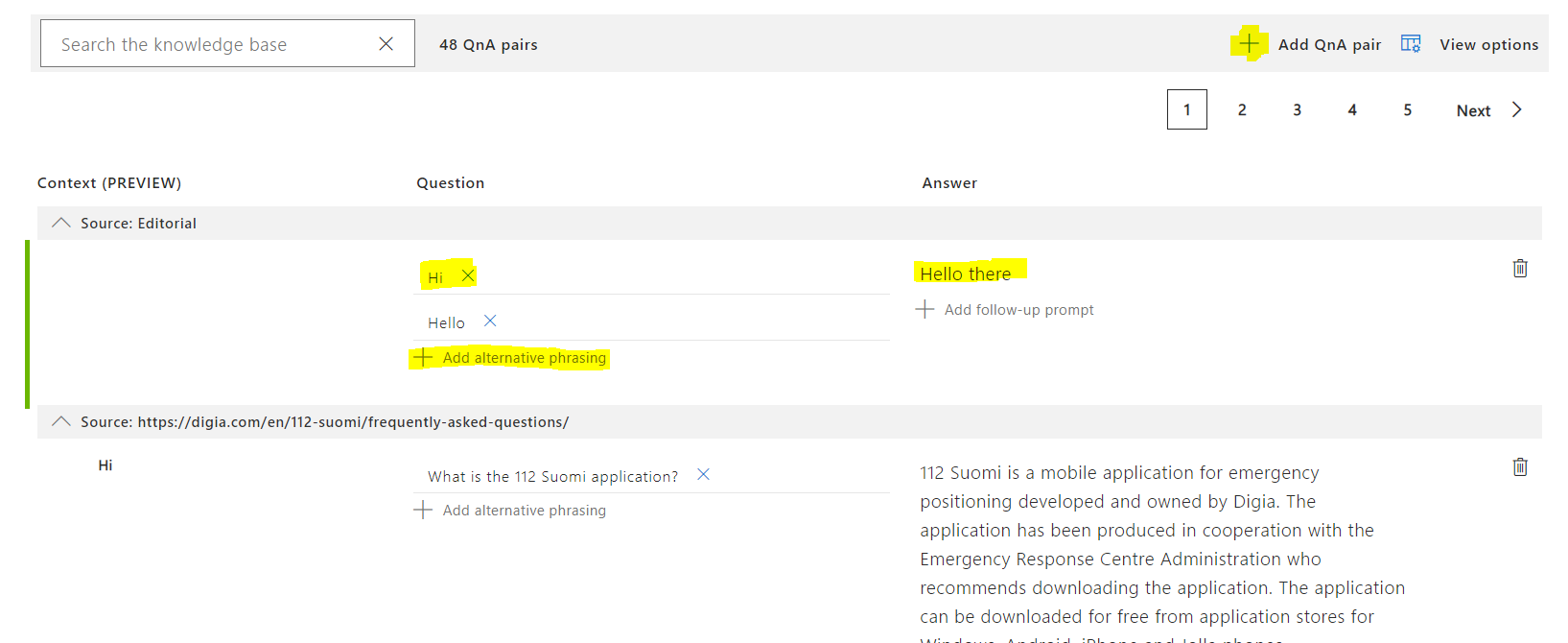
1. Chit-chat: None



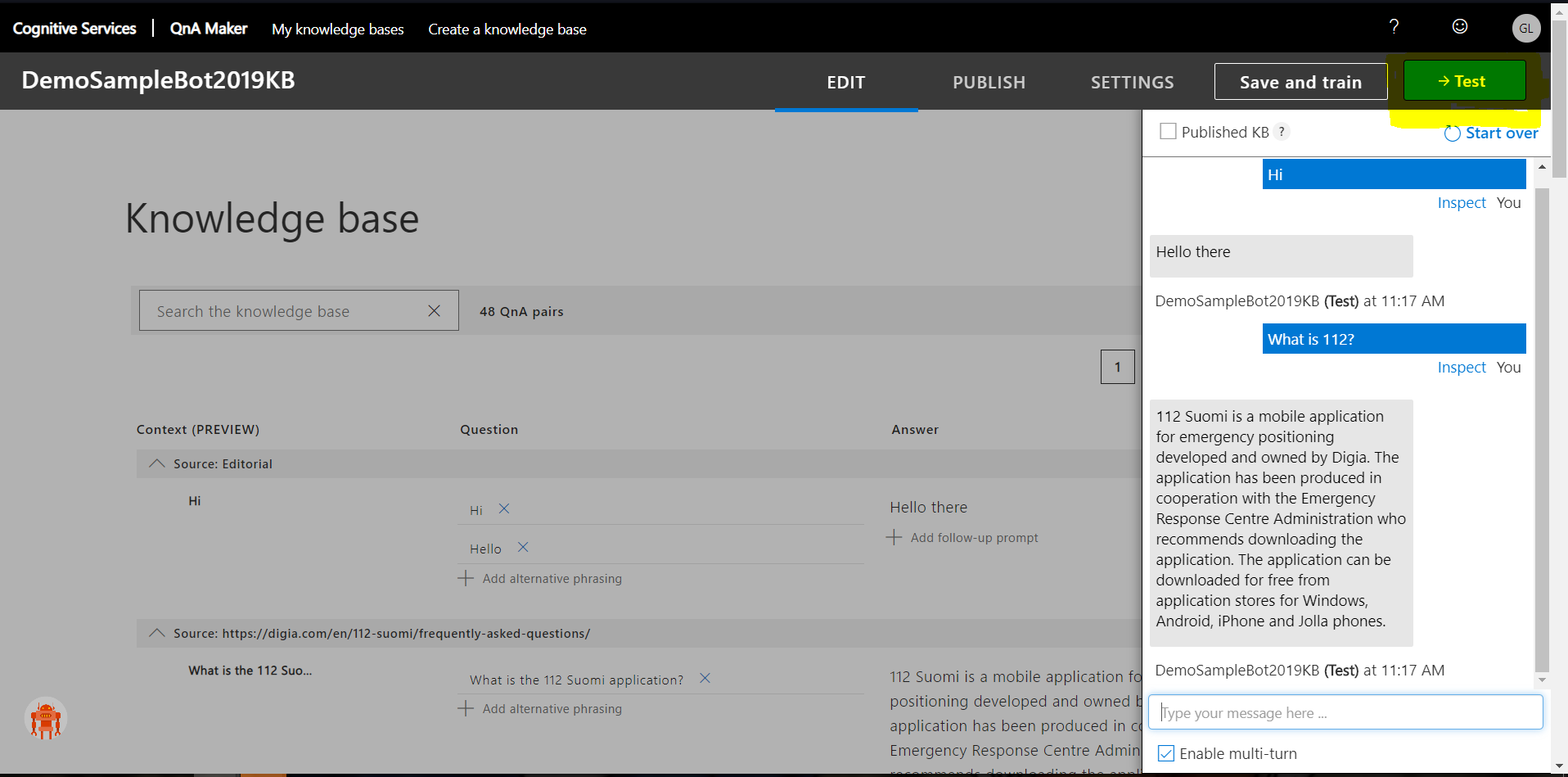
1. Click Create your kb to create the knowledge base. Once completed you will see a page like this.



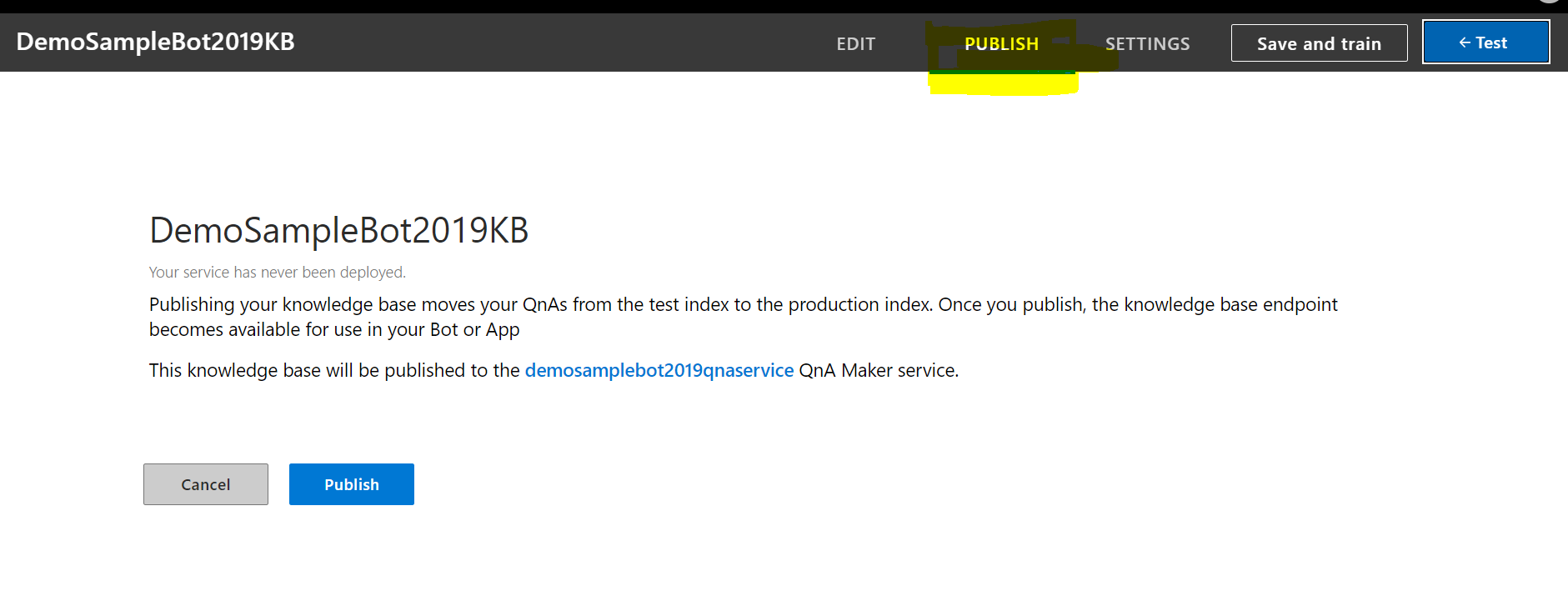
1. You can also add more QnA pairs by clicking Add QnA pair and adding Questions and Answers (Can also add alternate phrasing). Once done Save and train your knowledge base.



1. Then you can test it by clicking Test and validate your questions with their answers from the KB.

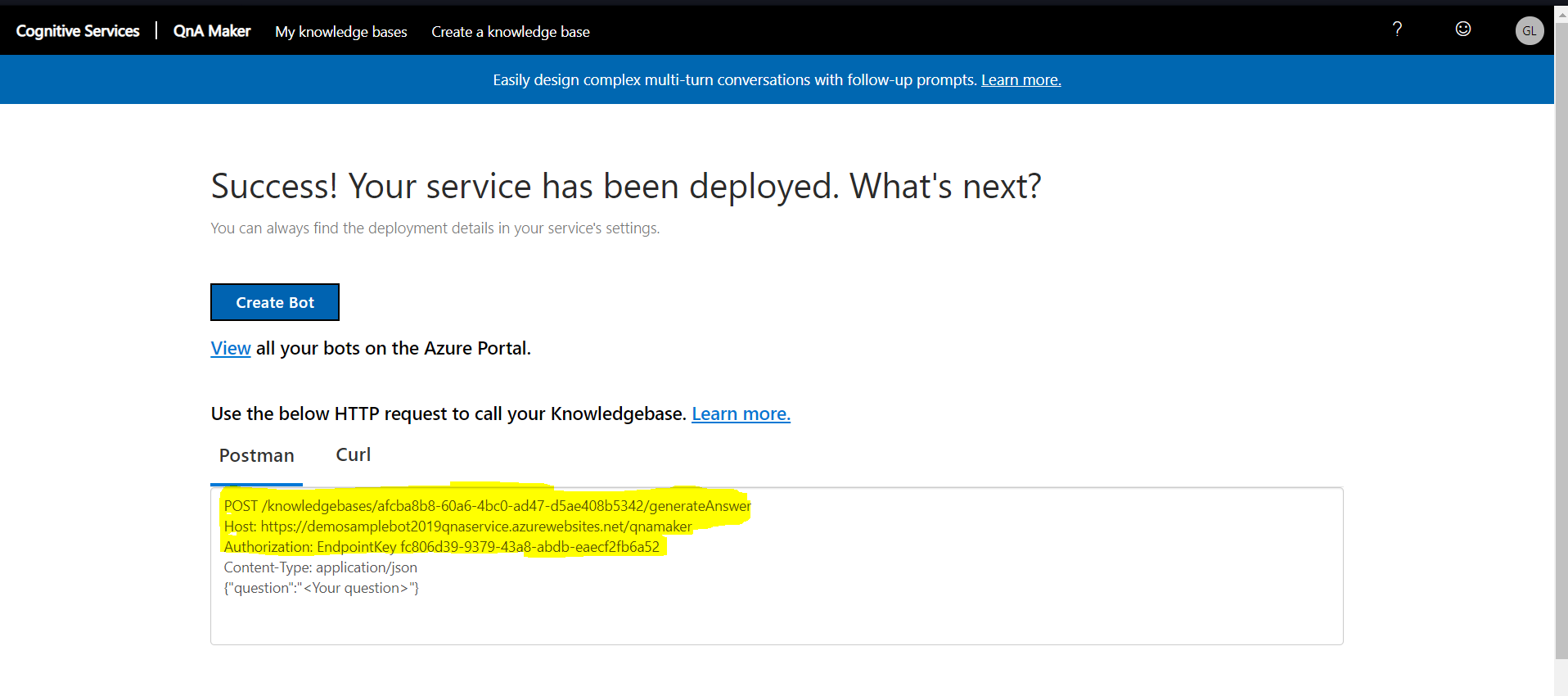


1. Publish your knowledge base.



1. After publishing, record the following values from deployment details

POST /knowledgebases/<knowledge-base-id>/generateAnswer  
Host: <your-hostname>// NOTE - this is a URL ending in /qnamaker  
Authorization: EndpointKey <qna-maker-resource-key>

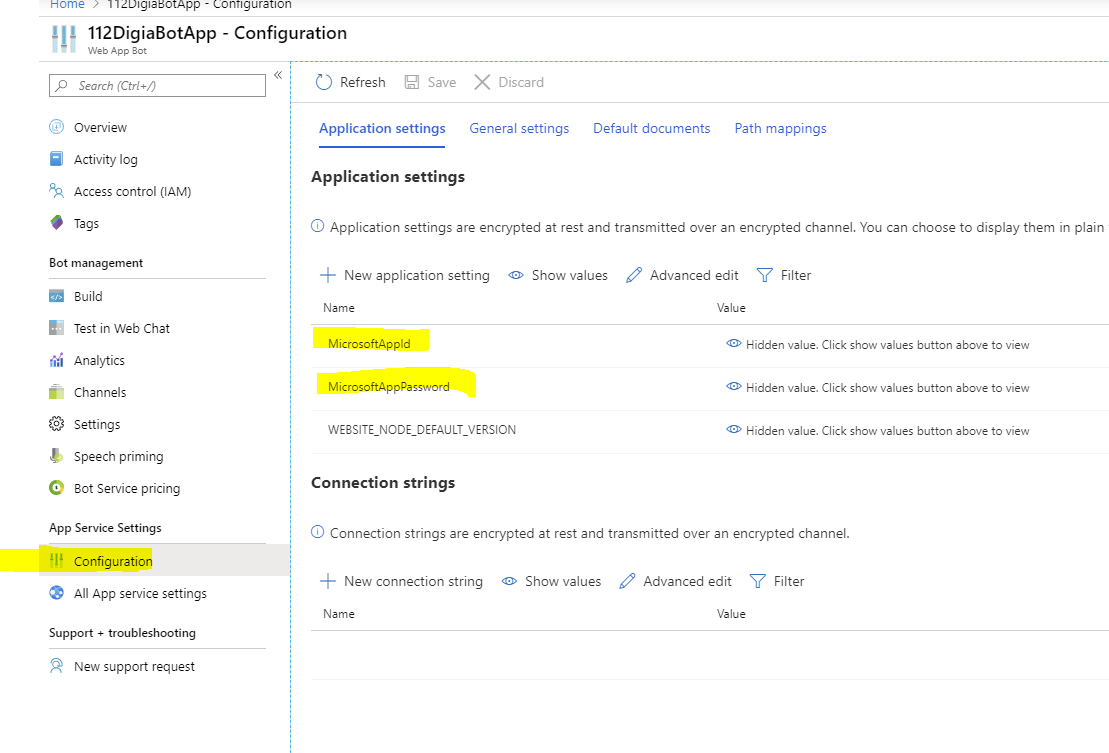


## Build 112 digia chat bot on top of QnA service:

Go to <https://github.com/SureshG02/QnAMaker_FAQs> and follow instruction written in README.md file.

**How to get MicrosoftAppId and MicrosoftAppPassword ?**

Go to Web App bot created in previous step and then go to Configuration under App Service Setting. Edit and copy them.



**How to get QnAKnowledgebaseId, QnAEndpointKey and QnAEndpointHostName ?**

Go to QnA KB you created in previous step and then go to Settings tab. You will find below details as Deployment Details:

POST /knowledgebases/ **<**QnAKnowledgebaseId> /generateAnswer

Host: <QnAEndpointHostName>

Authorization: EndpointKey <QnAEndpointKey>

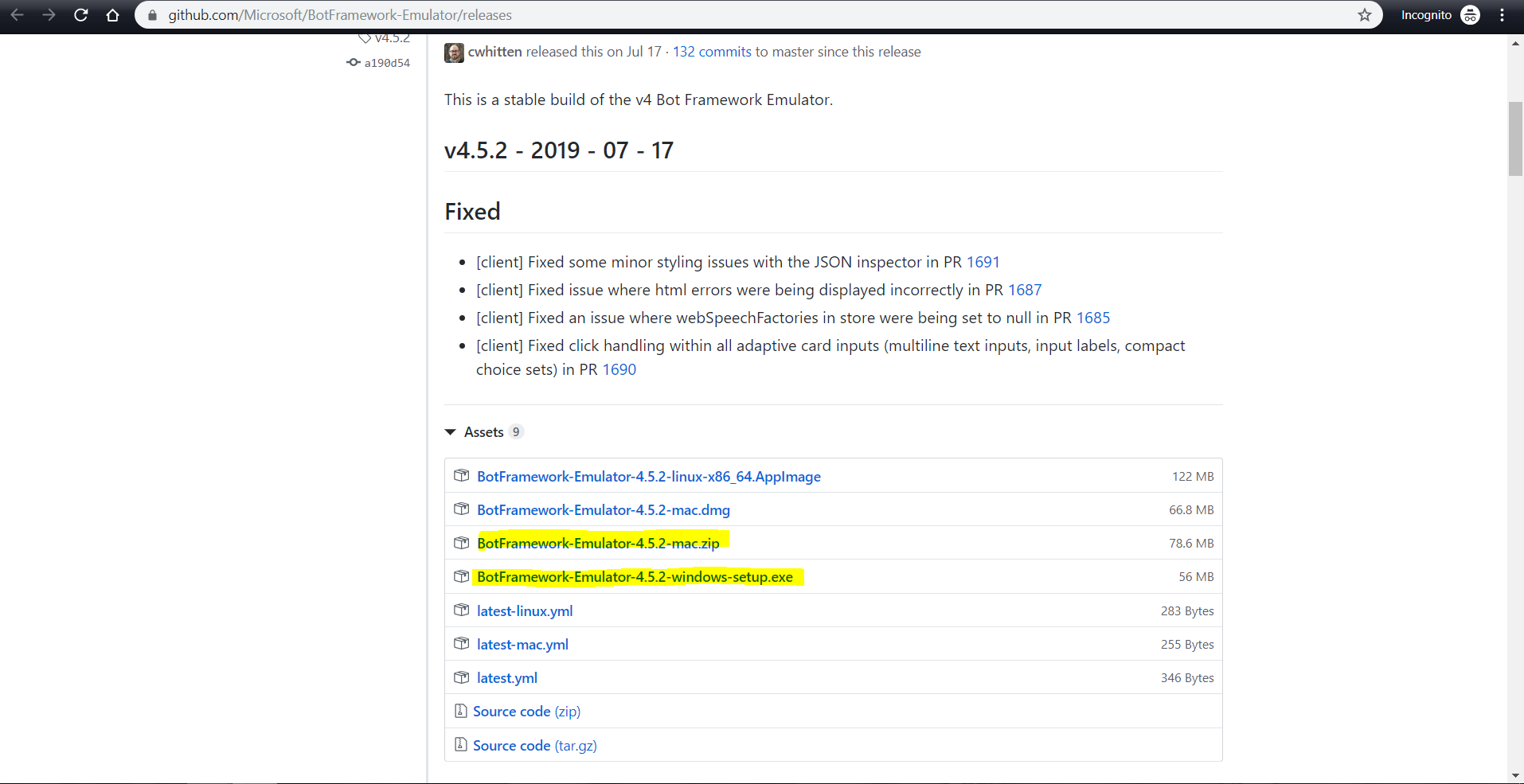
Content-Type: application/json

{"question":"<Your question>"}

**Test in local from Bot Emulator:**

1. Download emulator from the below link.

<https://github.com/Microsoft/BotFramework-Emulator/releases>



1. Open the Emulator and click File and New Bot Configuration.

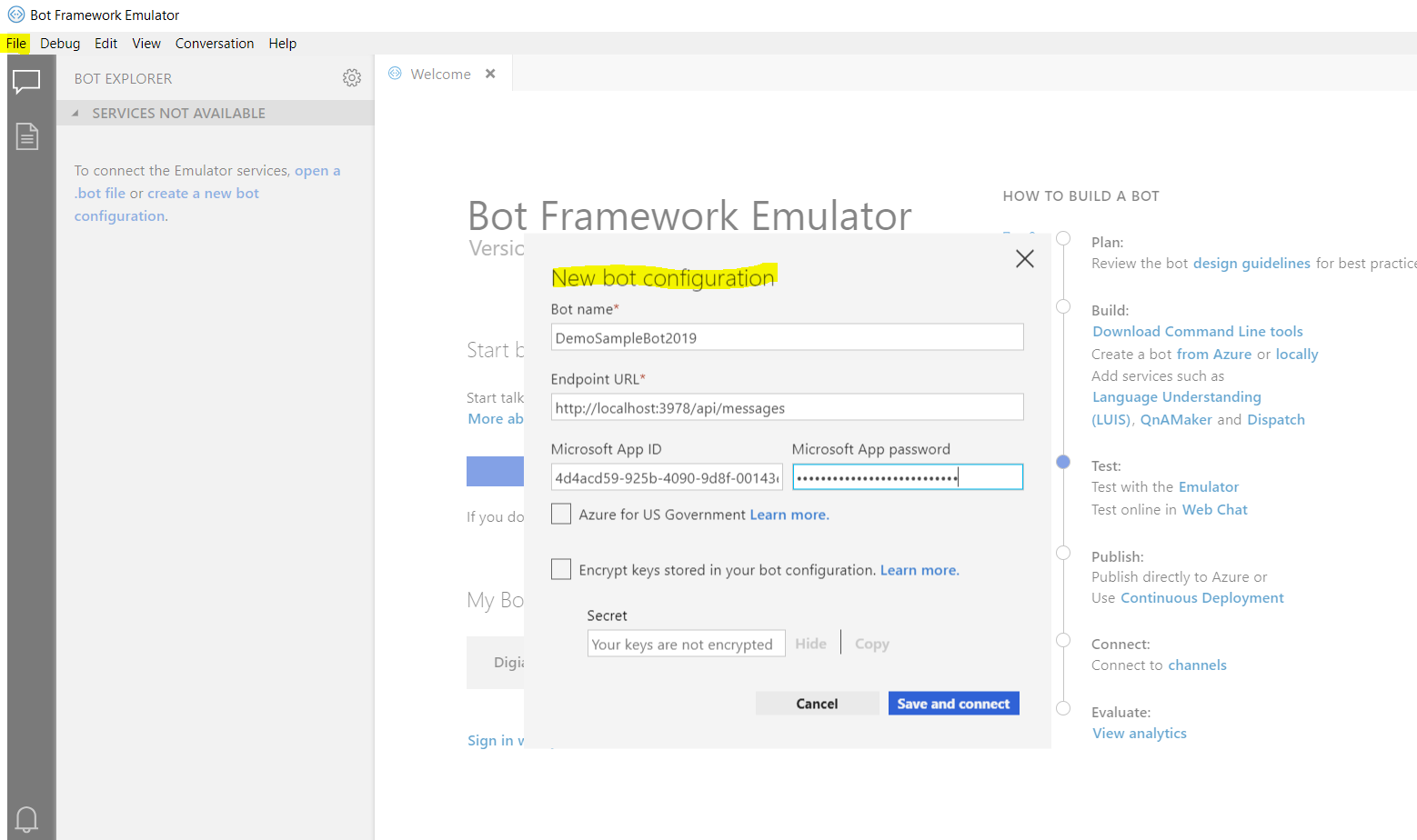
Name: Your Bot Name (Any name)

EndpointURL: **<http://localhost:3978/api/messages>**

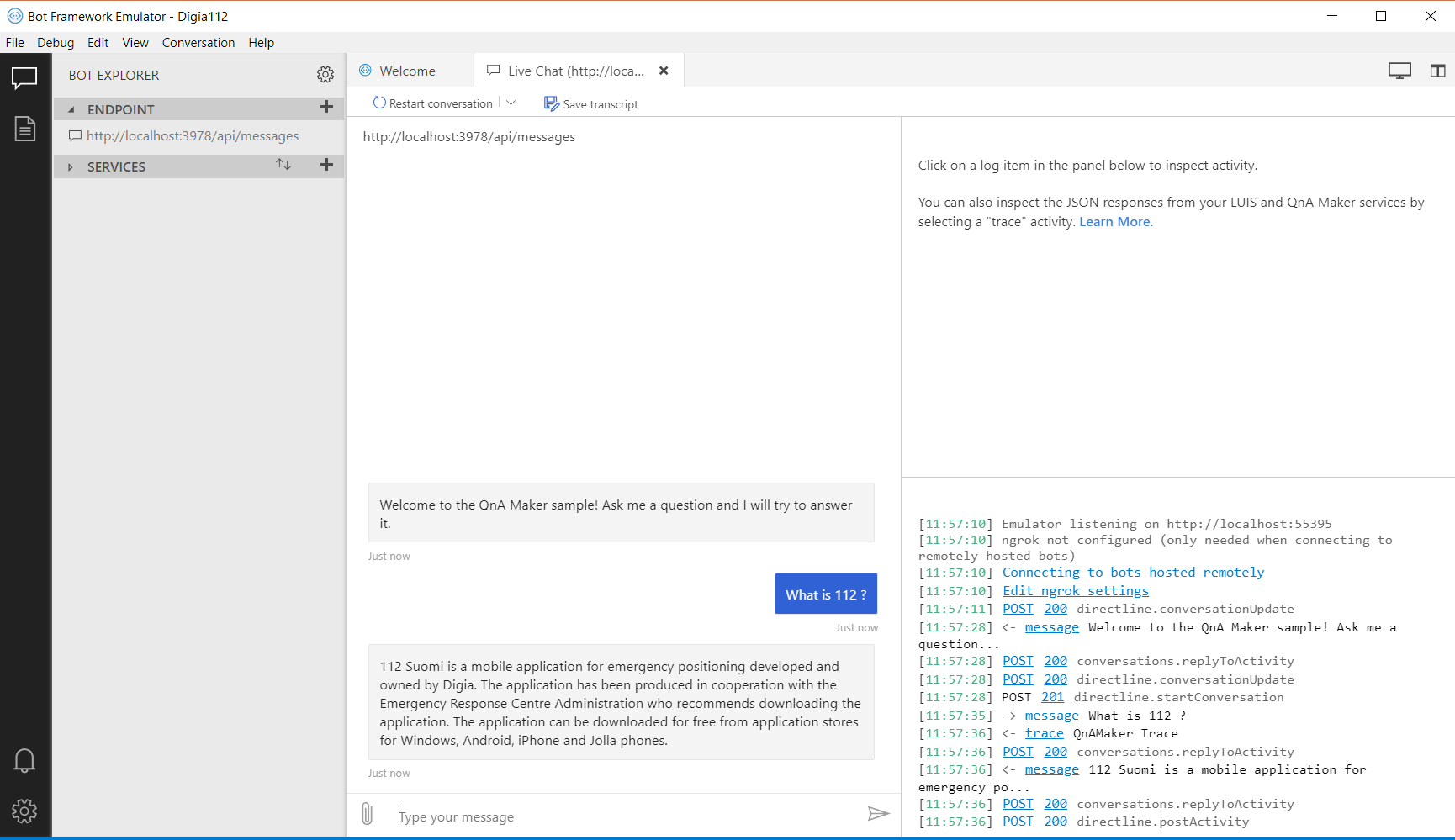
MicrosoftAppId: from .env file

MicrosoftAppPassword: from .env file

then click **Save and Connect**

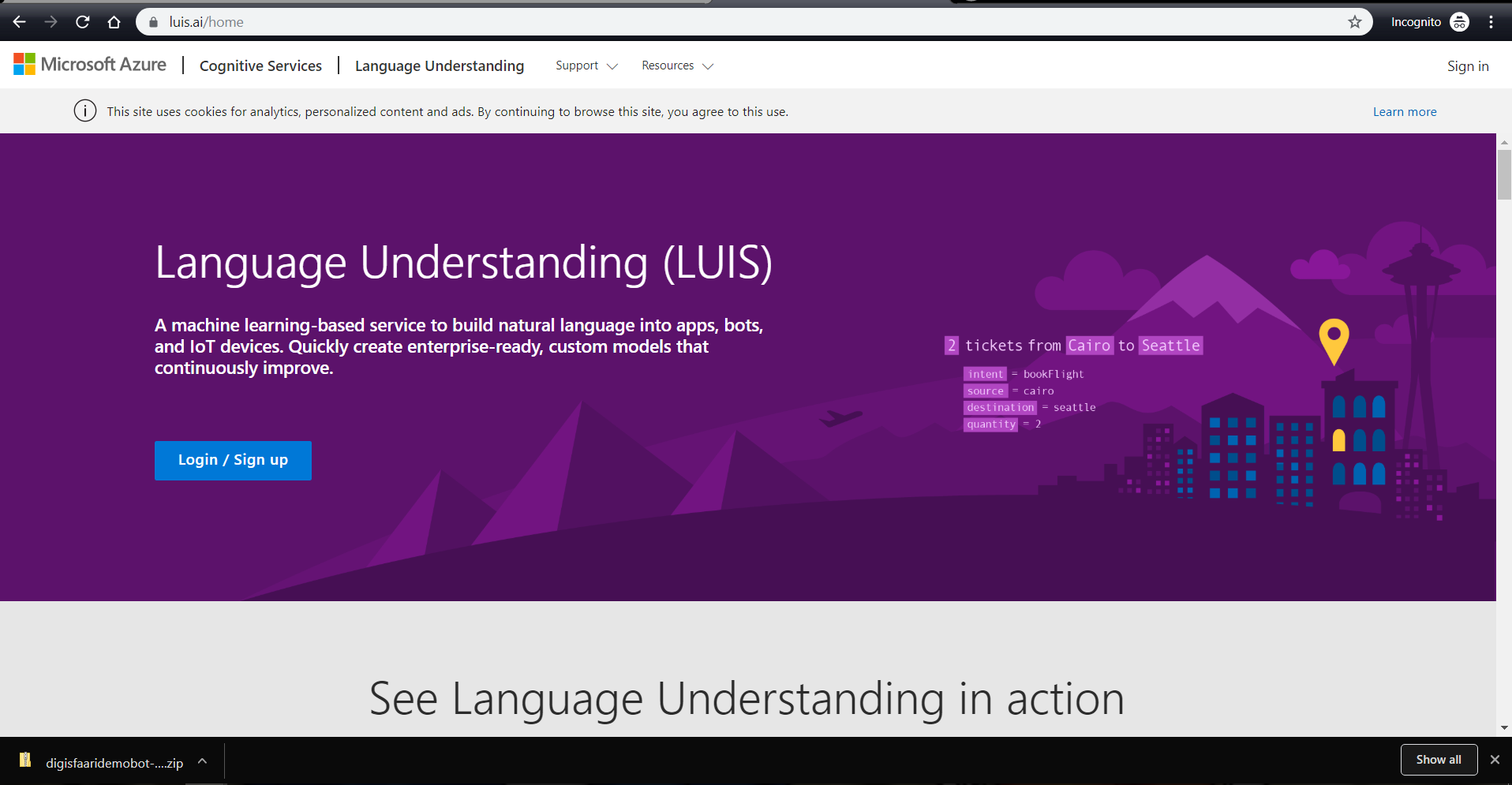


1. Once connected you can test your chatbot by asking questions from your KB

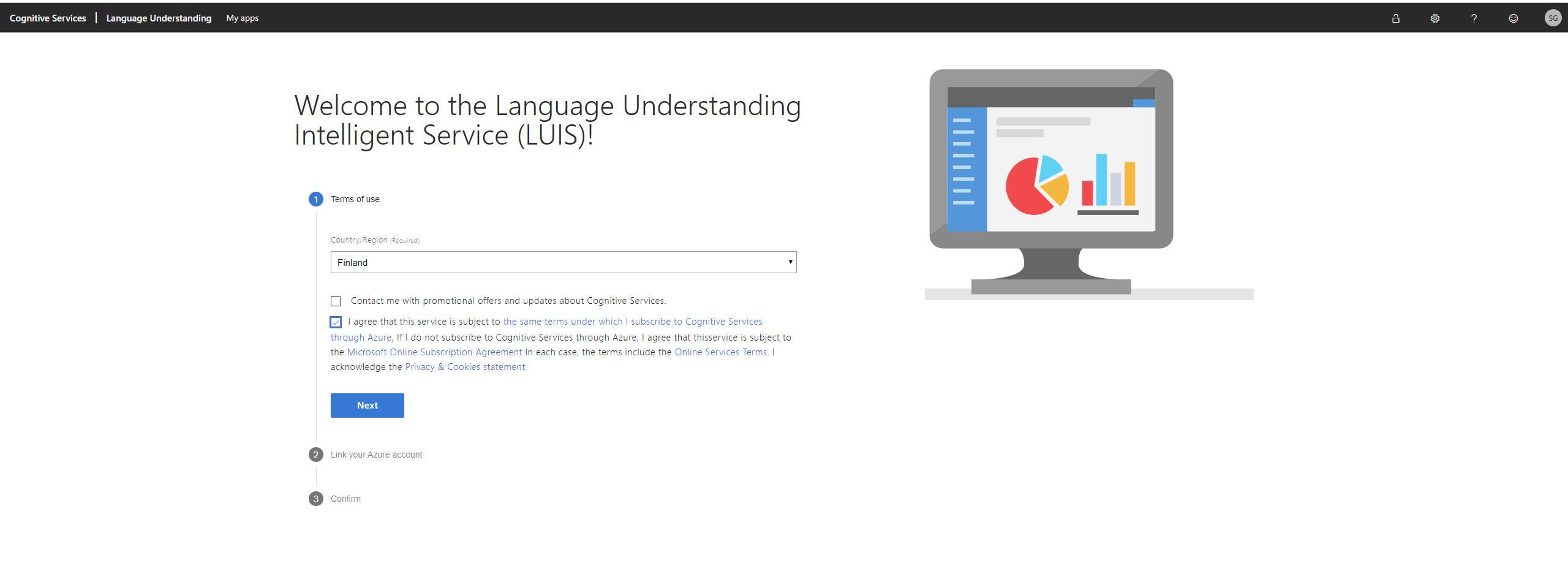


## Make 112 digia chat bot more intelligent using LUIS:

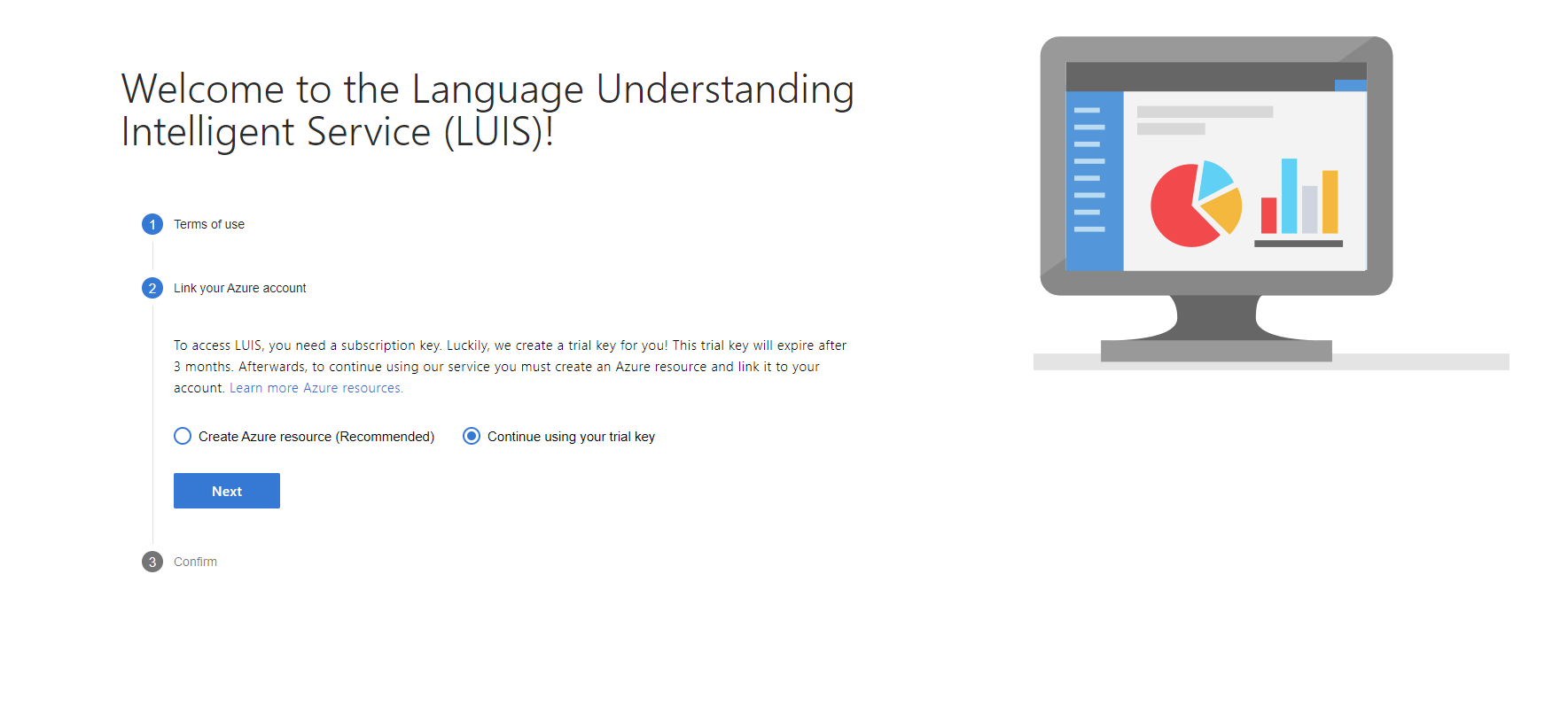
1. Go to <https://www.luis.ai/home> and sign in



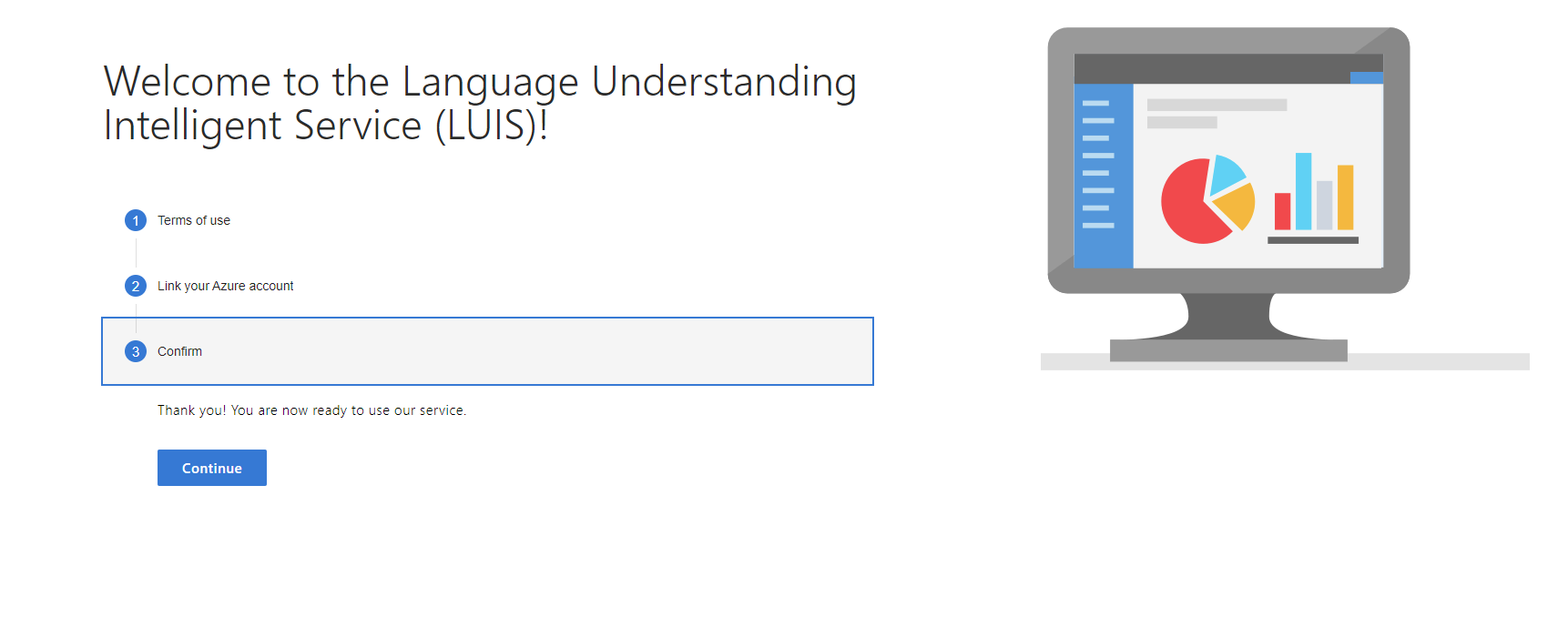
1. Choose country



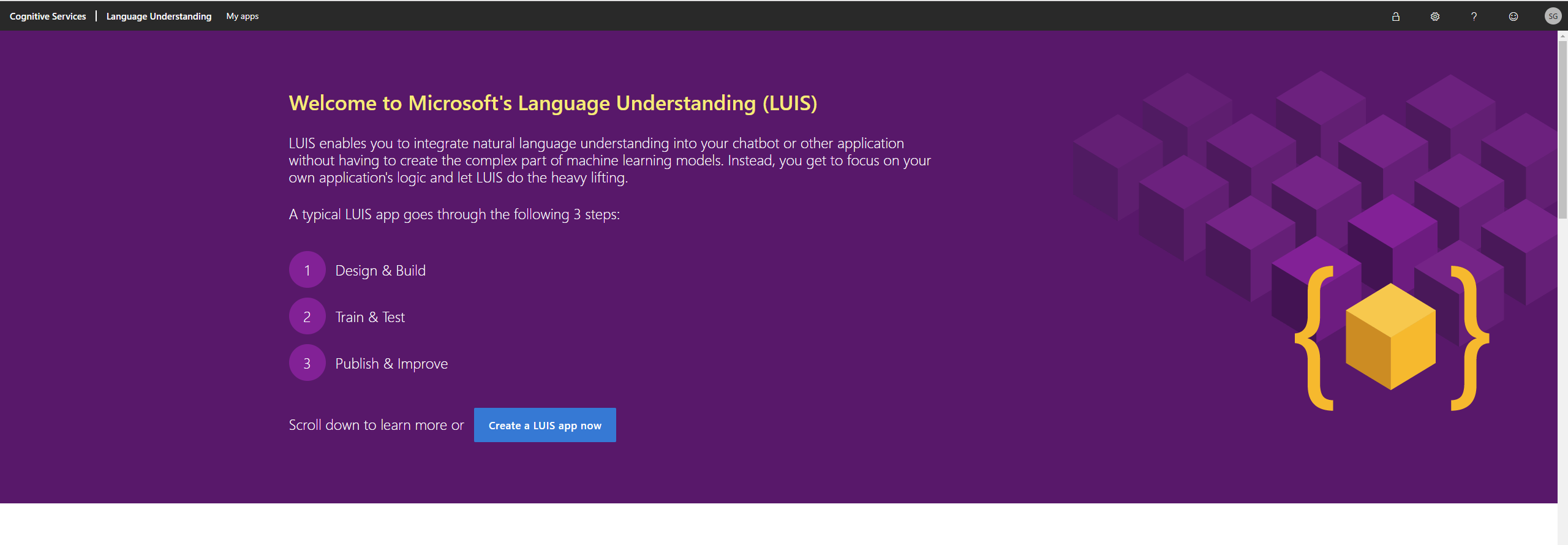
1. Continue using your trial key

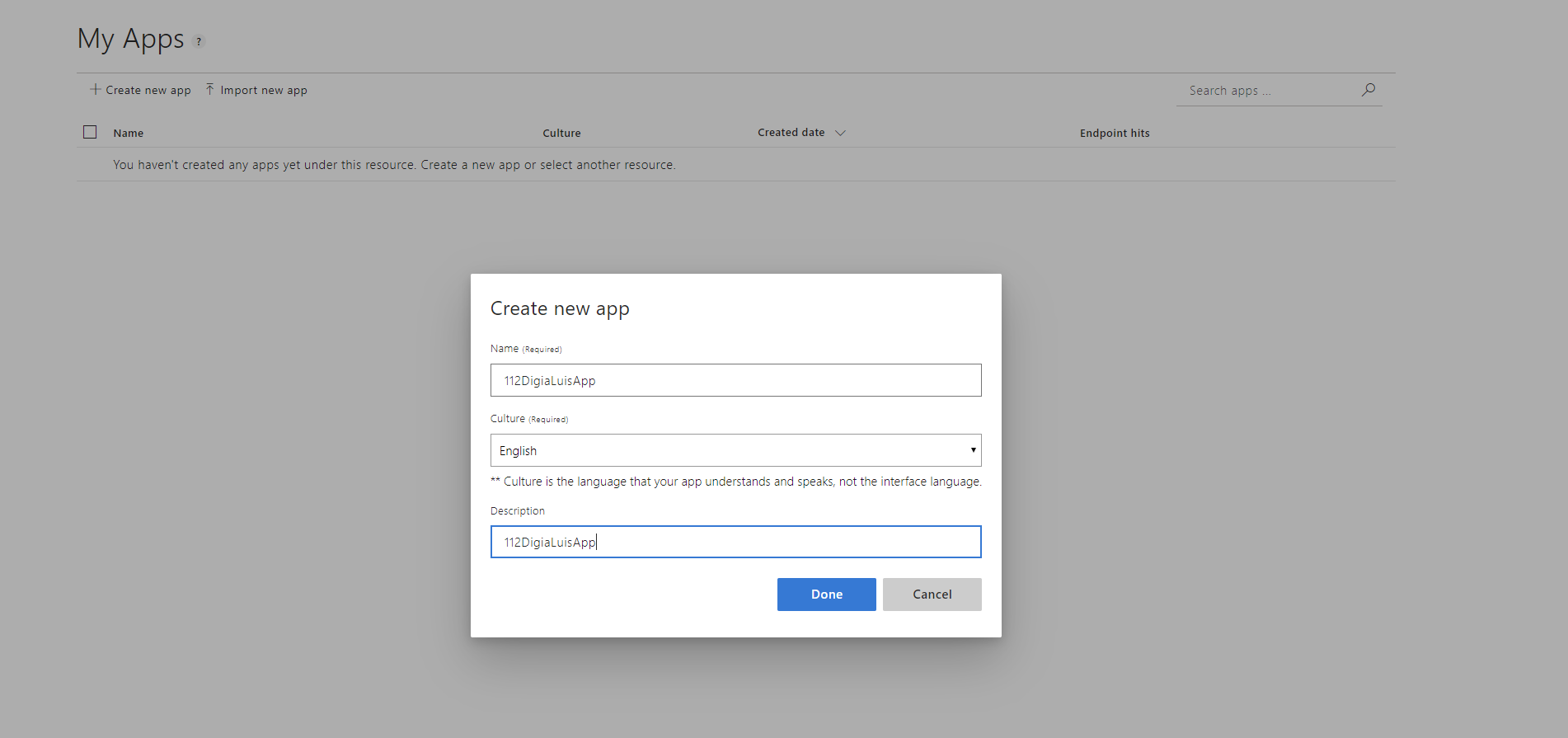


1. Confirm



1. Create LUIS app. Provide app name and Click Done.





Now go to <https://github.com/SureshG02/LUIS_FAQs> and follow steps mentioned in README file.

Add Intent, Utterance and Entities as mentioned in README file. Label utterances with required entities. Train and Publish your LUIS model.

**How to get LuisAppId, LuisAPIKey and LuisAPIHostName ?**

Open LUIS app you created. Go to MANAGE => Cope Application ID as LuisAppId

Go to Azure Resources and copy LuisAPIKey and LuisAPIHostName as shown below:

Primary key: < LuisAPIKey>

Endpoint Url: https://< LuisAPIHostName >/luis/api/v2.0