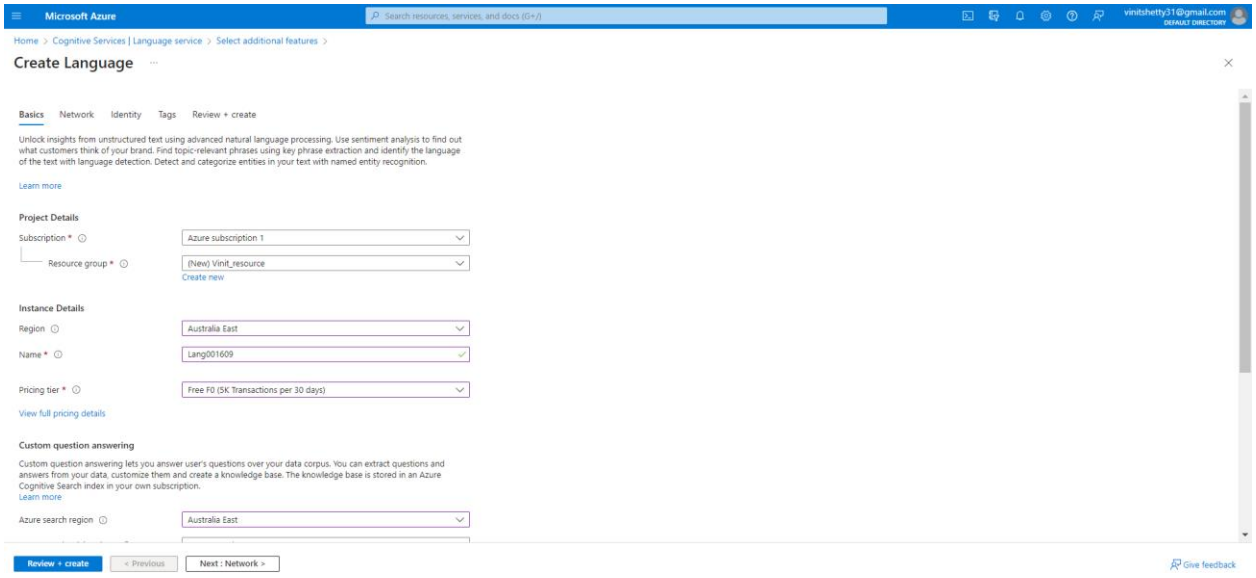
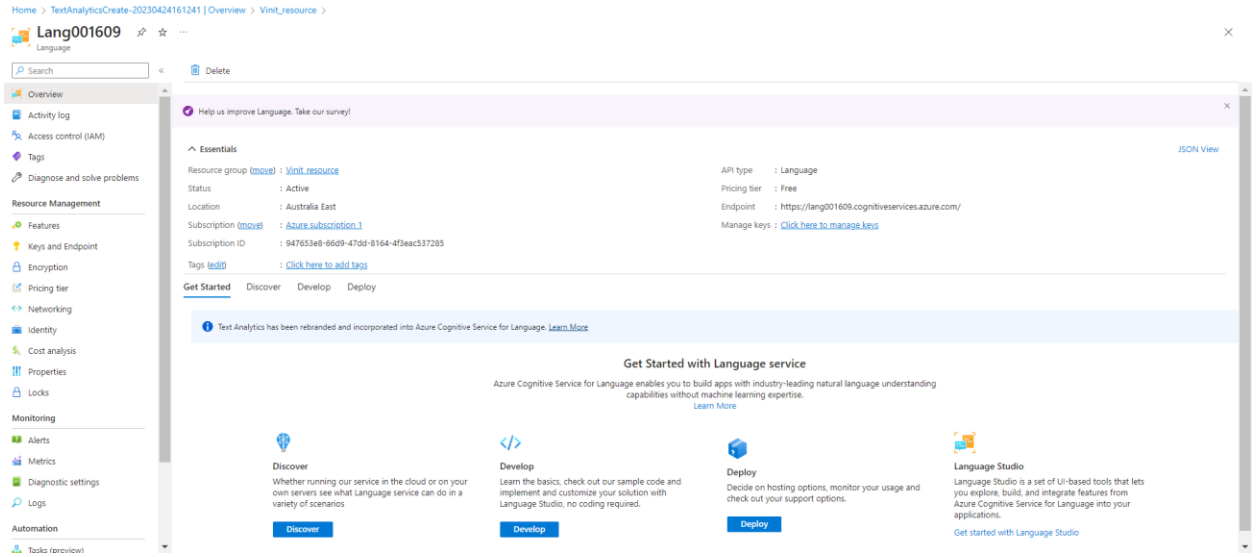
## 1. Creating Resources for Language Service

For creating a Custom Question Answering bot using Azure Cognitive Service Language Studio, we have created a new resource under the "**Language Services**" section. We choose the Subscription, Resource Group, Name, region, and Pricing Tier for the new resource and click on Create to set up a language resource for the project.

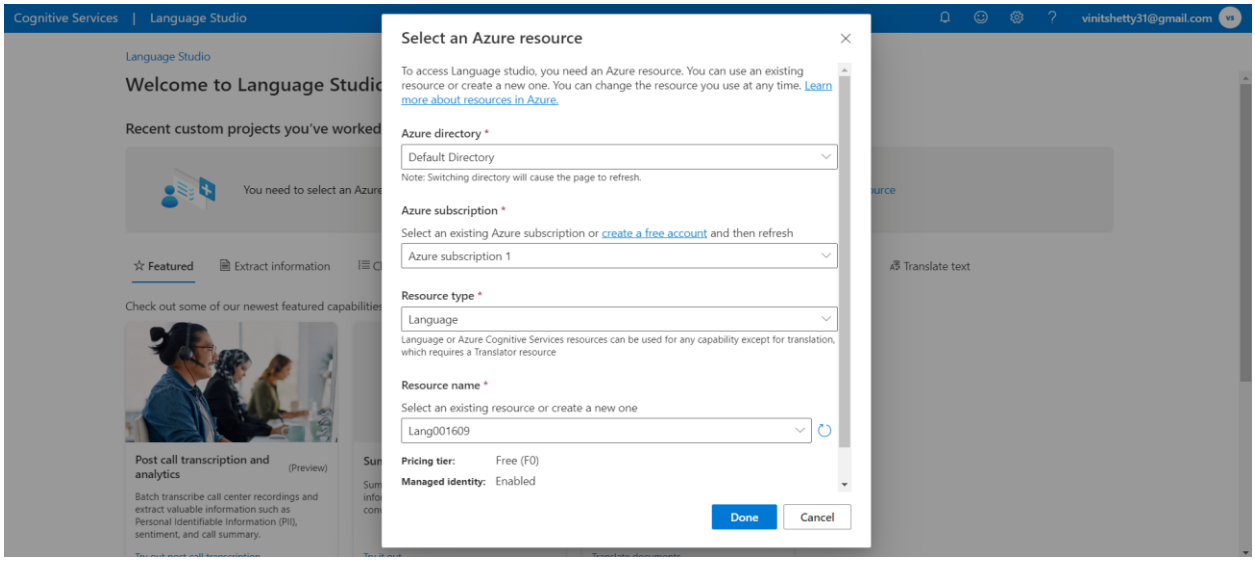
We can get the **API Keys**, **Endpoint URL** which can be used later to create the project in Language Studio and create the Bot.



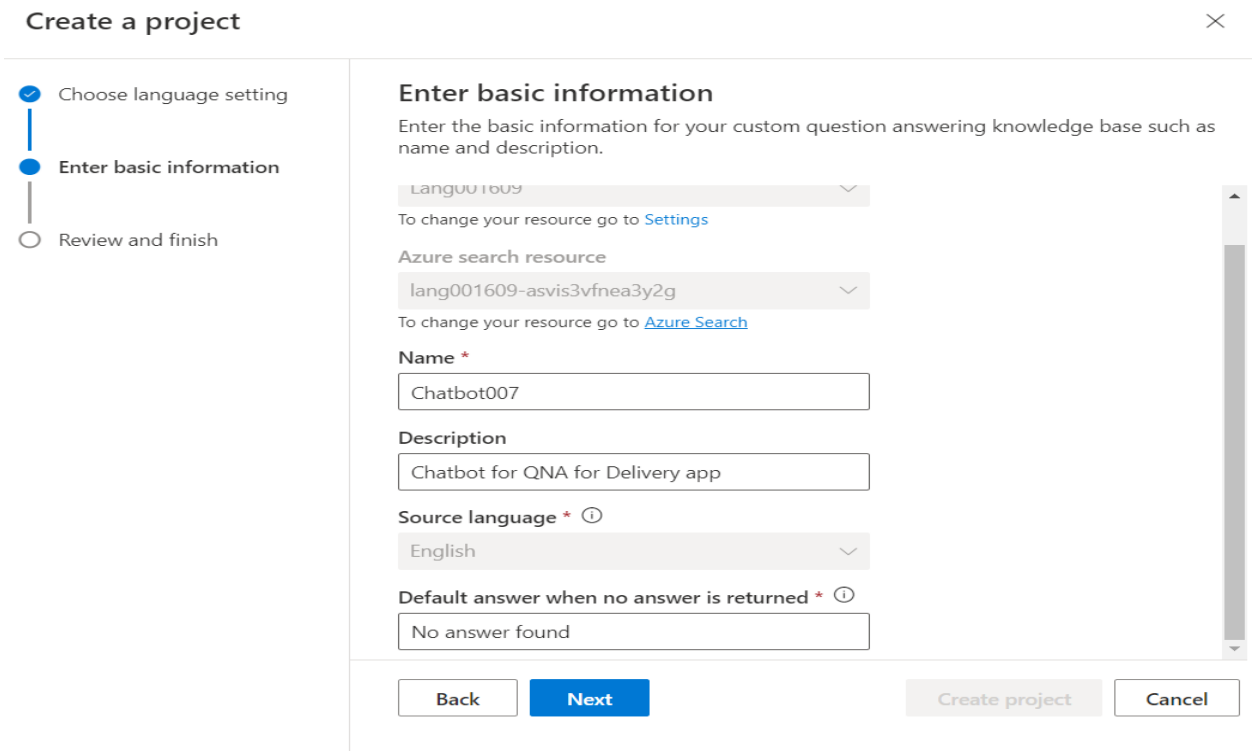
## 2. Selecting an Azure Resource in Language Studio

Once a Language Resource is generated in the Azure Portal, we proceed to access the language studio and choose the specific resource that was created earlier, namely "**Lang0011609**". Then, we navigate to the "**Understanding Questions and Conversational Language**" section and opt for "**Open Custom question answering**" to initiate the project creation process.



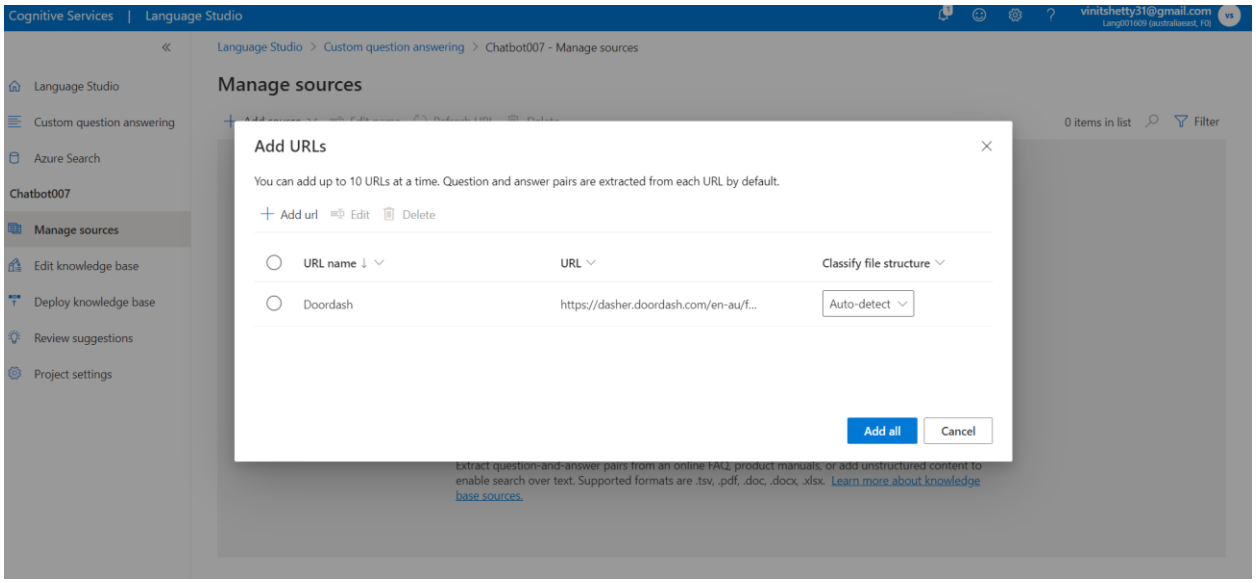

## 3. Creating a project

Once we have chosen the desired resource in the language studio, we proceed to click on the "**Create New Project**" option. Then, we select the preference of selecting the same language for all the projects within the resource and choose "**English**". Subsequently, we provide a suitable name for the project, such as "**Chatbot007**", furnish a brief description, and set the default answer as "**No Answer Found**". Finally, we click on the "**Create Project**" button to create the project.



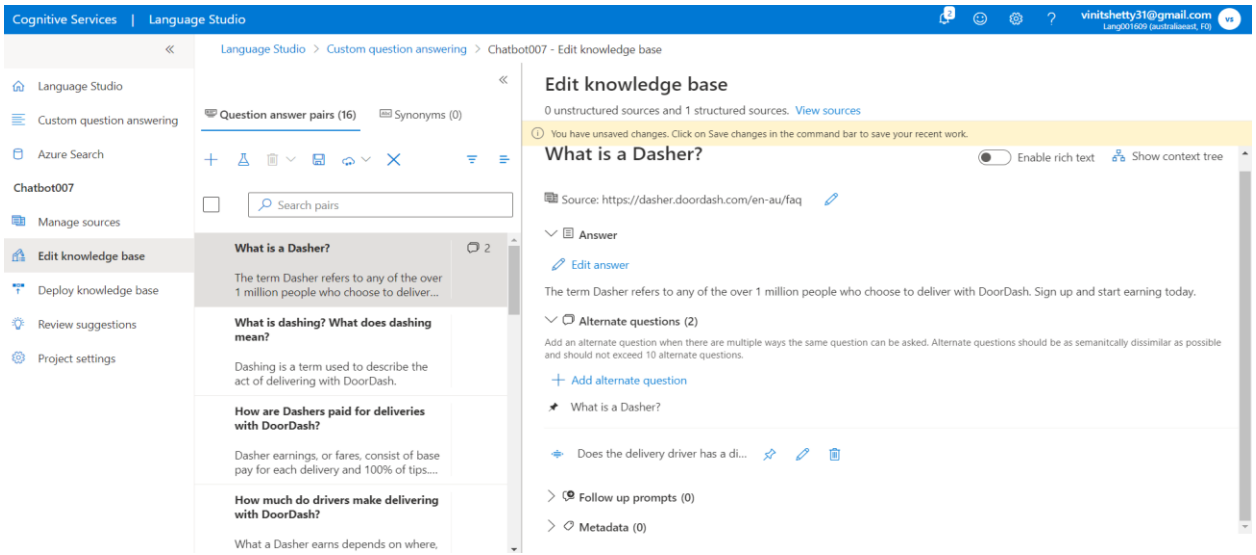
4. **Add a source from the chosen URL.**

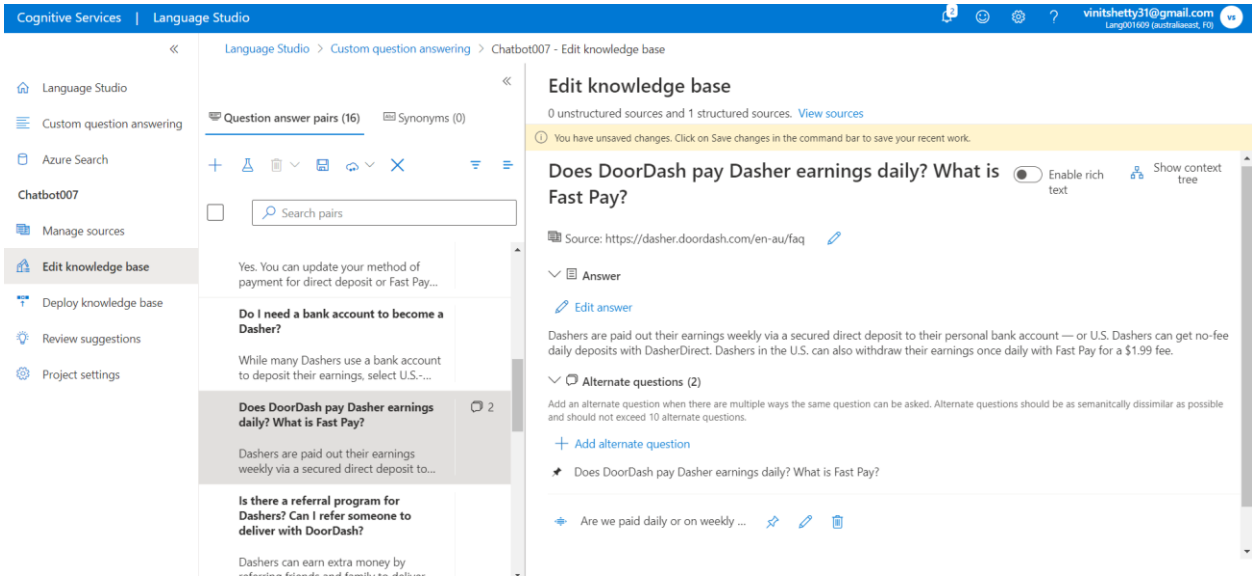
To add a fresh source, we need to access the "**Manage Sources**" tab and then click on the "**Add Source**" option, following which we select "**Add URLs**". For the current project that centres around "**Food Delivery Services**", we have chosen to utilize information from the **Doordash** FAQs. This website provides answers to some of the frequently asked questions by delivery partners, and its URL is [https://dasher.doordash.com/en-au/faq](https://dasher.doordash.com/en-au/faq)




5. **Editing the knowledge base with alternative questions.**

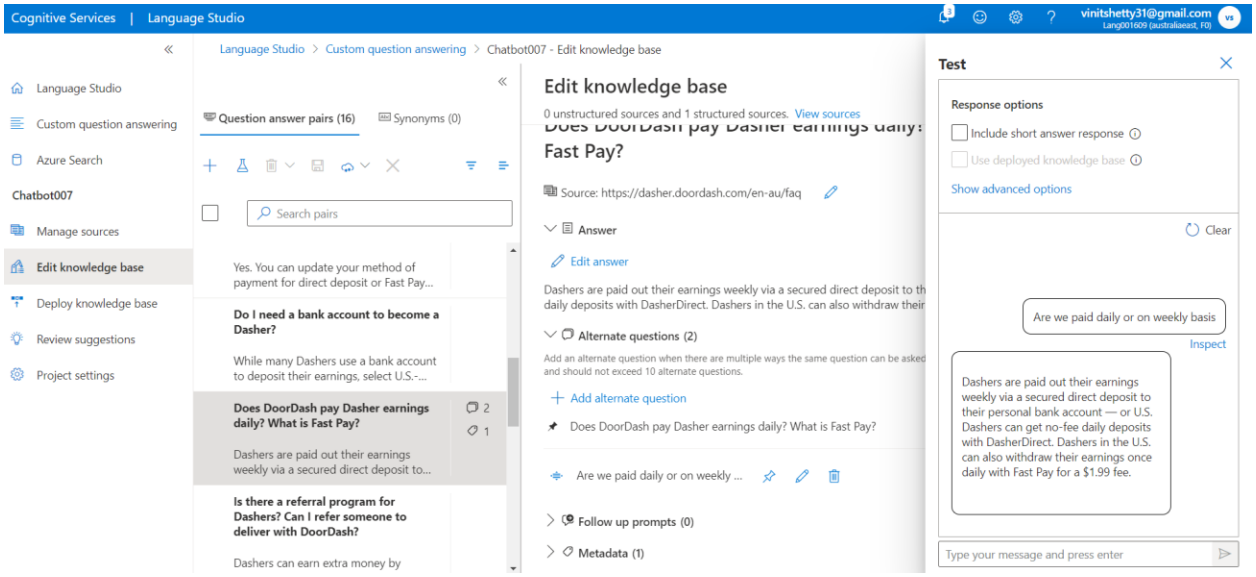
After we provide the source URL, it takes some time for the knowledge base to load. Once loaded, we can view suggested question-answer pairs that can be used as the bot's knowledge base. We also have the option to add more questions or alternate questions to improve the bot's understanding of user queries. For instance, an alternate question "**Are we paid daily or on weekly basis**" was included to enhance the original query "**Does Doordash Pay Dasher earnings daily? What is Fast Pay?**" The inclusion of alternate questions helps to enhance the knowledge base, enabling the bot to provide more accurate answers.





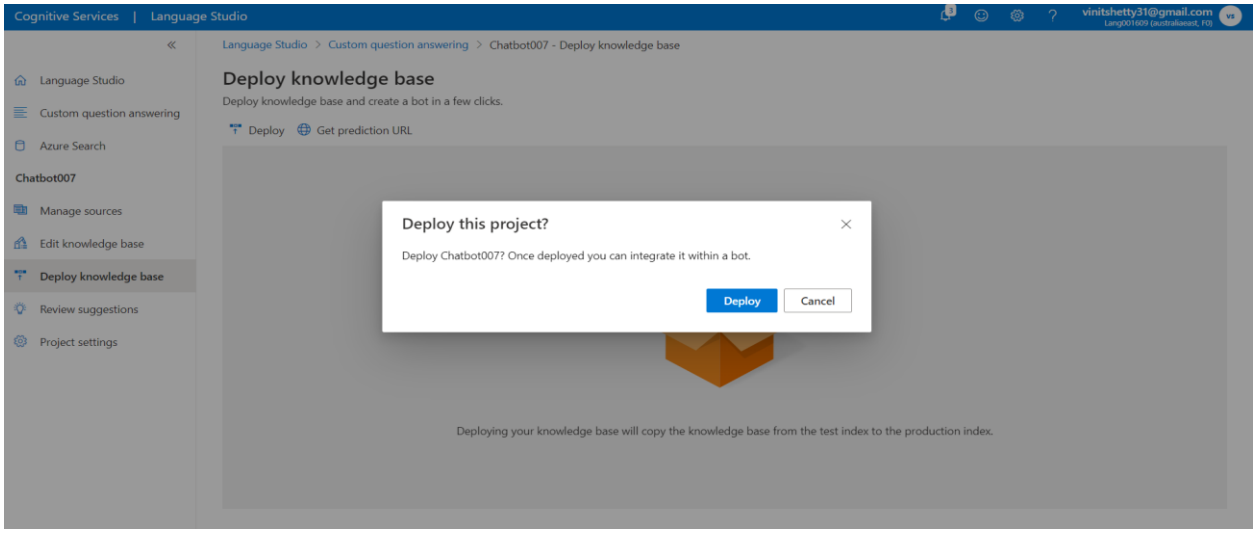

## 6. Testing the Project

Once we have updated the knowledge base with alternate questions, we can proceed to test the accuracy of the knowledge base in providing answers. To do this, we select the "**Test**" option from the menu in the edit knowledge base tab, which opens a pop-up window. In this window, we can ask questions, such as "**Are we paid daily or on weekly basis**," which was an **alternate question** added in the previous tab. Based on the established knowledge base, an appropriate answer is generated to the question asked.




## 7. Deploying the Project

After testing the project's accuracy, the subsequent step is to deploy it by selecting the "**Deploy knowledge base**" tab and clicking the "**Deploy**" option. We must then confirm the deployment by clicking the "**Deploy**" option again. Once the project has been deployed, we can proceed with the creation of the **QnA bot**.



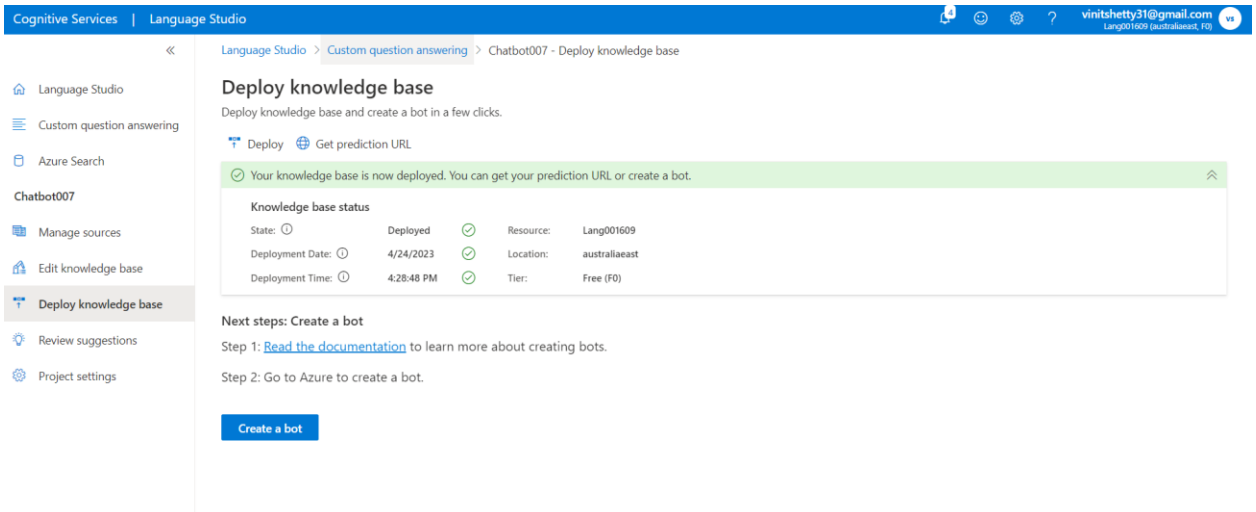

8. **Creating a bot**

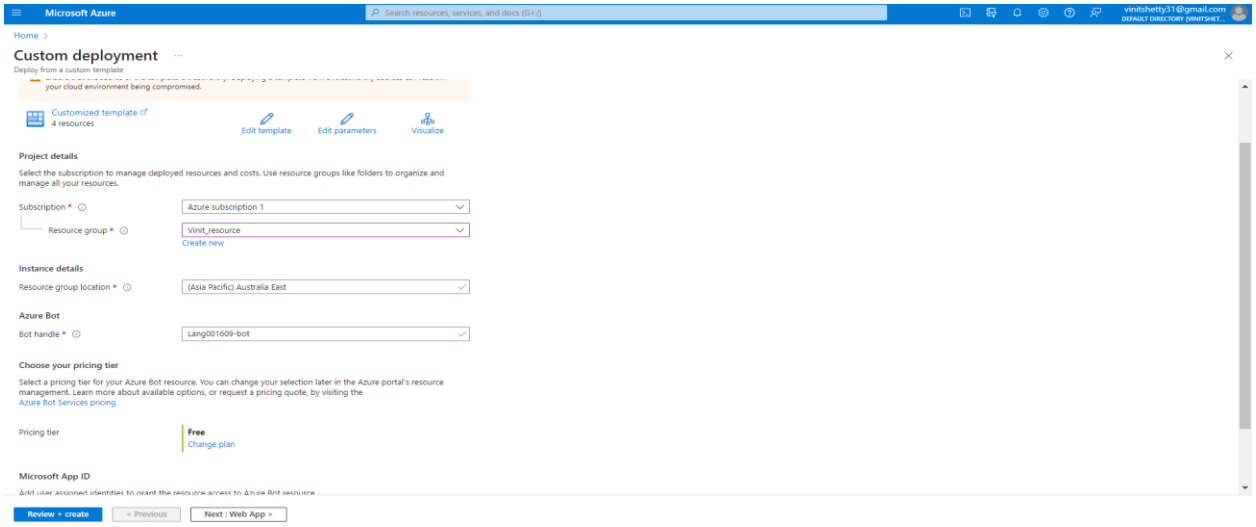
After deploying the project, we click on "**Create a bot**" which opens another browser tab in the Azure portal to create and deploy the bot.

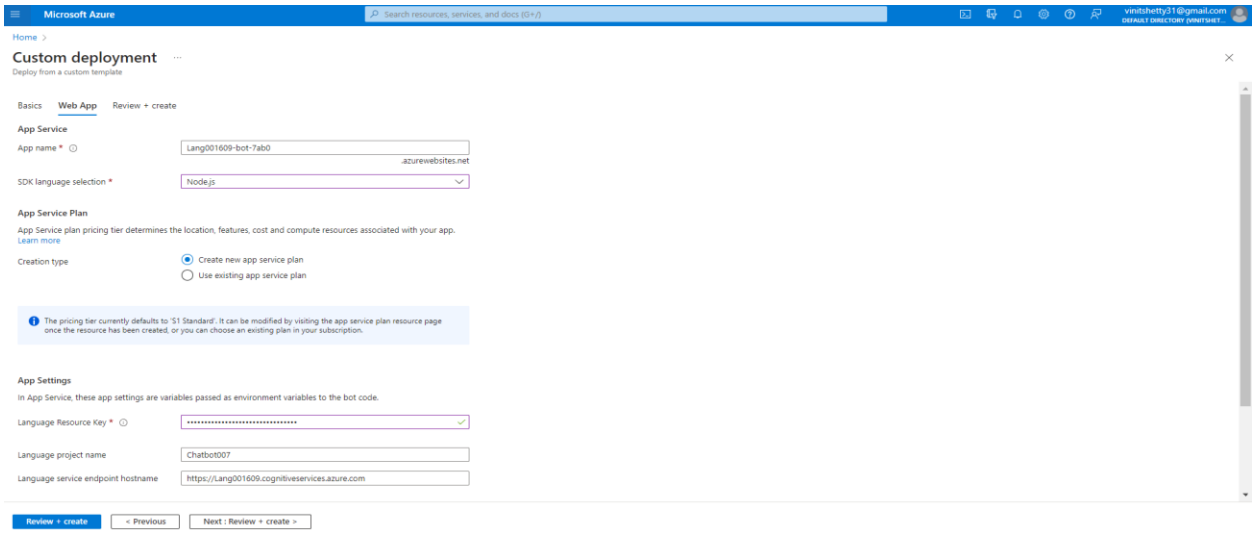
For creating the bot, we need to **configure** some values which are as below:

| **Setting** | **Value** |
|---|---|
| **Bot handle** | Lang0011609-bot |
| **Subscription** | Default Value |
| **Resource group** | Default Value |
| **Location** | Australia East |
| **Pricing tier** | Free |
| **App name** | Default Value |
| **SDK language** | Node.js. (JSON file) |
| **Language Resource Key** | Key obtained from the Language resource created |
| **App service plan/Location** | Default Value |

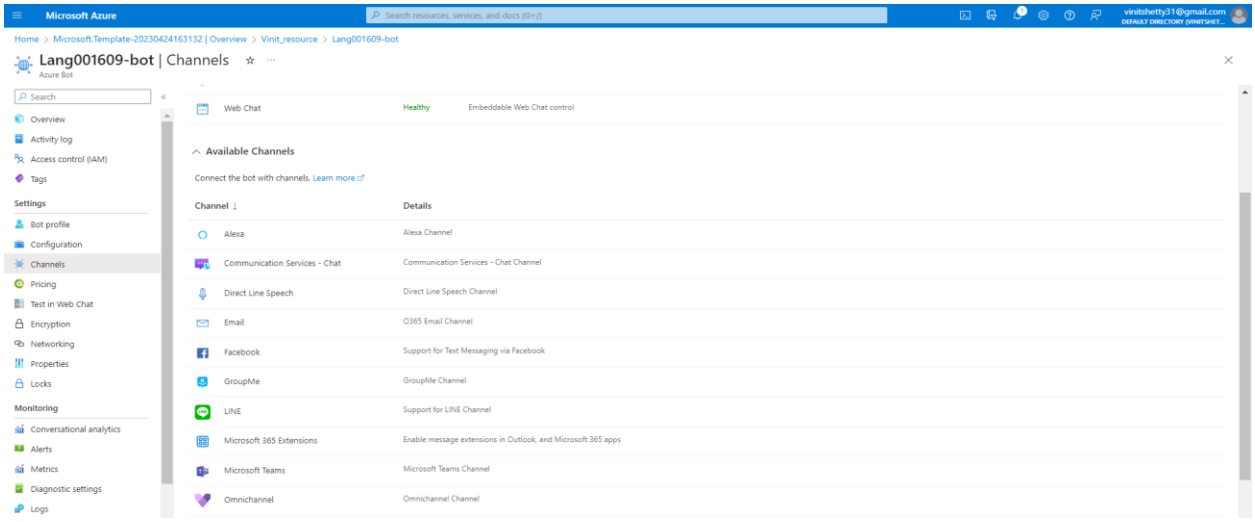
Upon configuring the necessary values, we click on "**Review+Create**" If the validation process is successful, the bot "**Lang0011609-bot**" is then created.






9. **Connecting the bot to the Microsoft Teams channel**
   Once the bot is successfully created, we can connect it to various channels by choosing the "**Channels**" tab within the create bot resource. In our project, we have integrated the bot with the "**Microsoft Teams**" channel, allowing us to directly ask the bot **Doordash Delivery agent-related queries** via Microsoft Teams.



## 10. Asking questions to bot on Microsoft Teams

Once the bot is successfully integrated into Microsoft Teams, we can begin asking it domain-specific questions. For instance, we asked the bot the question "**Do I get paid on a daily or weekly basis?**" to which the bot responded with "**Weekly**" and provided a brief explanation. Additionally, we posed a question to the bot that was **not present** in the knowledge base, and the bot returned the default response of "**No Answers found**."