

# Deakin University

## SIG788- OnTrack Submission

### Task 6.1 P

#### Submitted by

Neethu Sidhardhan  
S 223494027  
Attempt # 1  
28/4/2023

---

**Target Grade:** P

#### Task Details -

In this assessment, you need to provide your learning reflection by building a bot hosting on Python that can detect personal identifiable information.

To complete this you need to follow the instruction in the provided link and watch the recording of week 6 recordings.

This task will help you to understand how to create a bot using Azure bot framework SDK.

#### Steps Related in creating bot using Azure bot framework

- Login to Azure Portal and create a resource group

The screenshot shows the Microsoft Azure portal homepage. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and user account information. Below the navigation bar, there's a section titled "Azure services" with icons for "Create a resource", "All resources", "Cost Management ...", "Education", "Marketplace", "Subscriptions", "Resource groups", "Help + support", and "Azure Machine Learning". A blue arrow points from the "Create a resource" icon towards the "Recent" tab in the "Resources" section. The "Resources" section has tabs for "Recent" (which is selected) and "Favorite". It lists four resources: "SIG788-task5" (Resource group, last viewed a week ago), "Free Trial" (Subscription, last viewed 2 weeks ago), "Azure for Students" (Subscription, last viewed 2 weeks ago), and "DefaultWorkspace-centralindia" (Log Analytics workspace, last viewed a month ago). At the bottom of the resources list, there's a "See all" link and a URL "https://portal.azure.com/#create/hub".

Fig 1 Azure Portal

➤ Create a new resource group

The screenshot shows the Microsoft Azure Resource groups page. At the top, there is a search bar and a navigation bar with options like 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. Below the search bar, there are filter buttons for 'Subscription equals Free Trial' and 'Location equals all'. A table lists three resource groups:

Name	Subscription	Location
DefaultResourceGroup-centralindia	Free Trial	Central India
NetworkWatcherRG	Free Trial	Central India
SIG788-task5	Free Trial	East US

At the bottom, there are navigation buttons for 'Page 1 of 1' and a 'Give feedback' link.

Fig 2 Create resource group

➤ Fill the form for creating resource group

The screenshot shows the 'Create a resource group' form in the 'Basics' tab. The form has three tabs: 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab contains the following fields:

- Project details:**
  - Subscription: Free Trial
  - Resource group: (empty input field)
- Resource details:**
  - Region: (US) East US

At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Tags >'.

Fig 3 Create a resource group - Form

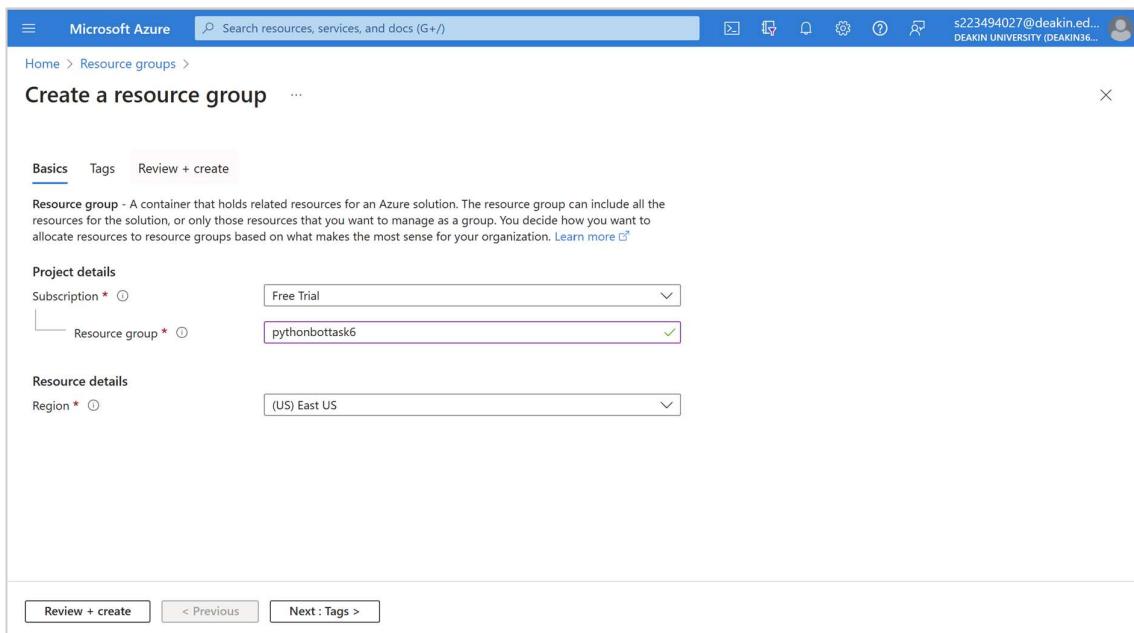


Fig 4 Create resource group

#### ➤ Validation process for the resource group

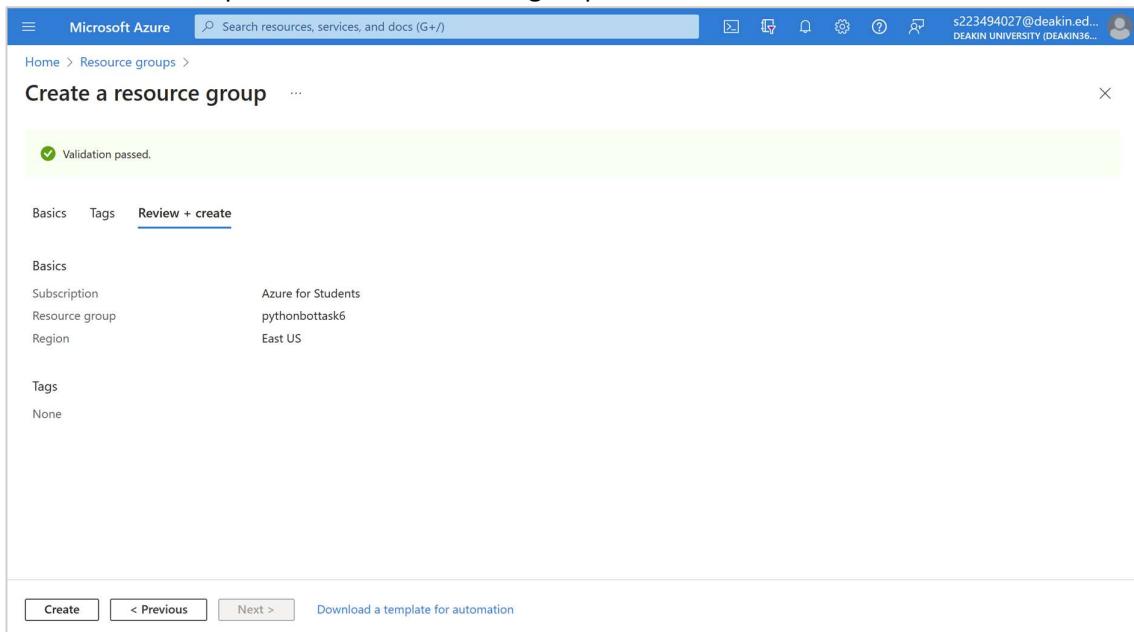


Fig 5 Validation of resource group

- Resource group is created and listed under the resource group list.

Name	Subscription	Location
DefaultResourceGroup-centralindia	Free Trial	Central India
NetworkWatcherRG	Free Trial	Central India
pythonbottask6	Azure for Students	East US
SIG788-task5	Free Trial	East US

Fig 6 Resource group list

- Overview of the resource group created.

Fig 7 Overview of resource group

- Now, we need to create language services. For getting the services, select the option “create”. The window assigns to “Market Place” where we get all the services from there.

- Market place where we can create different services.

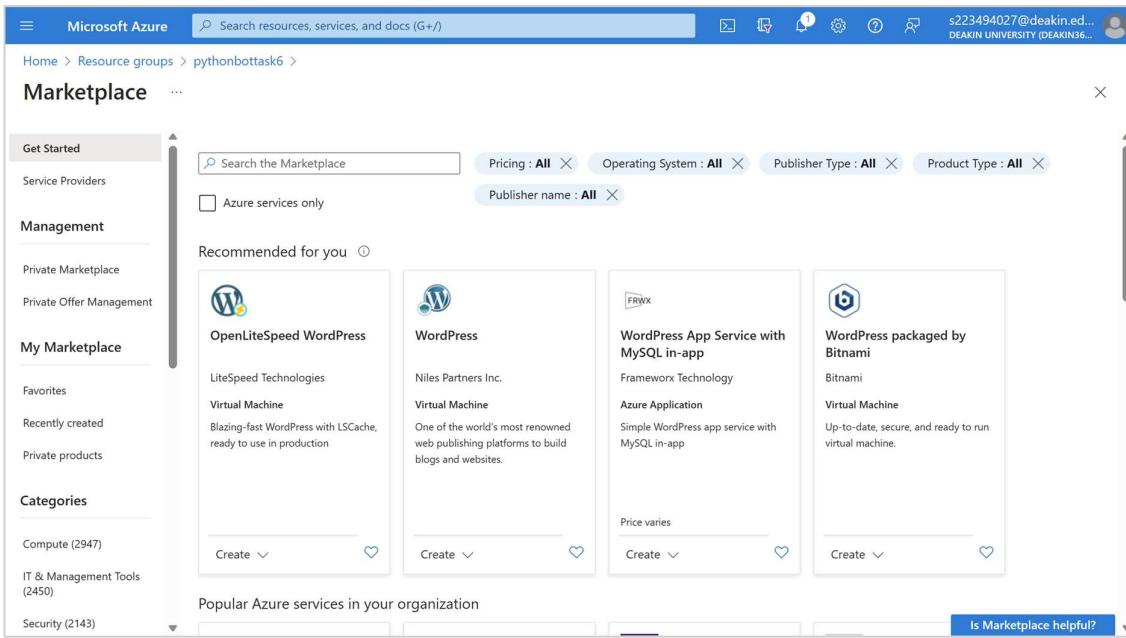


Fig 8 Market place

- Search for “Language Services” for building text analysis related to bot creation.

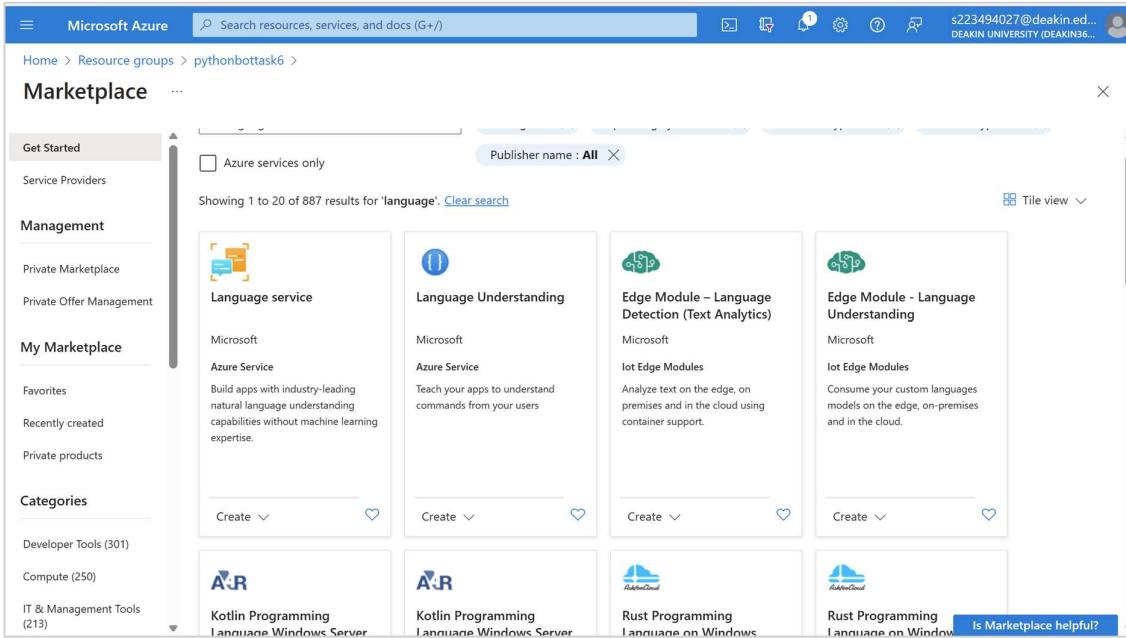


Fig 9 Language Services

- Select the option “Create” in language services. Here, we get window where we need to select the additional features like custom question answering, text classification and so on. Here, in this step we are not selecting any of the features.

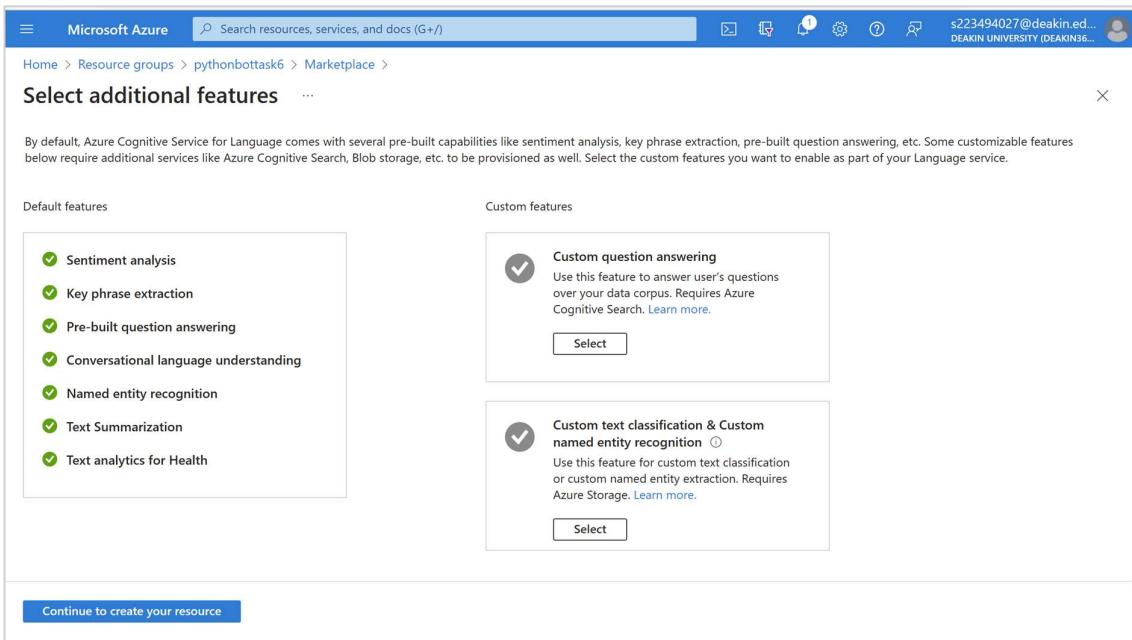


Fig 10 Select additional features

➤ Create languages under the resource group that we created.

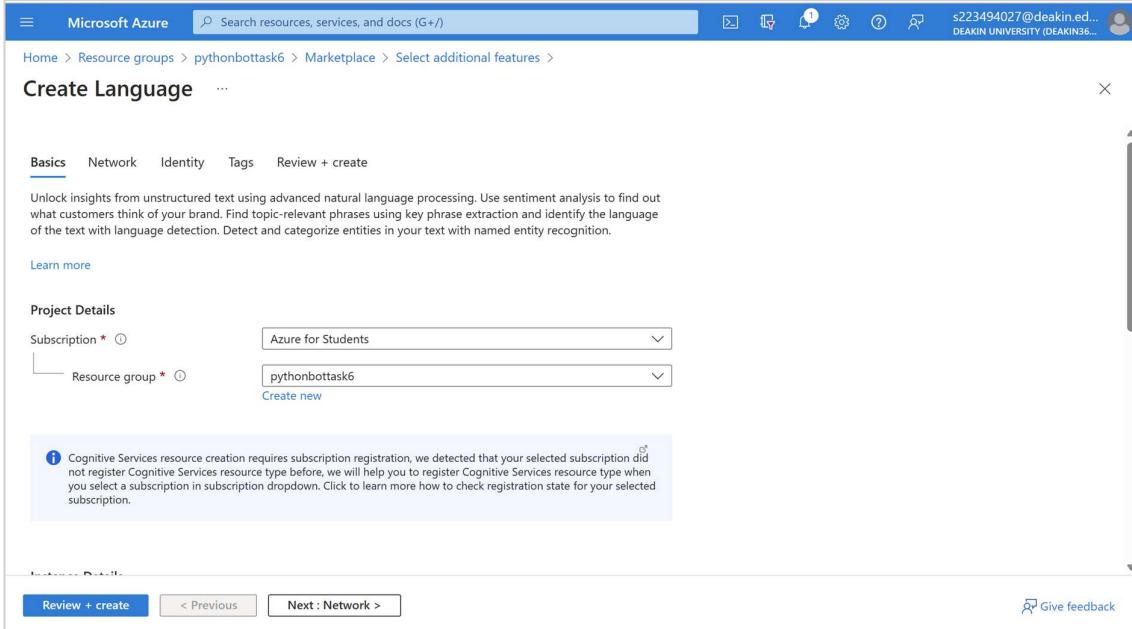


Fig 11 Create Language

- While creating the language, we need to read and select the three documents related to responsible AI. Once the documentation is selected, select the option create, where the system will run a validation process and the language group is created.

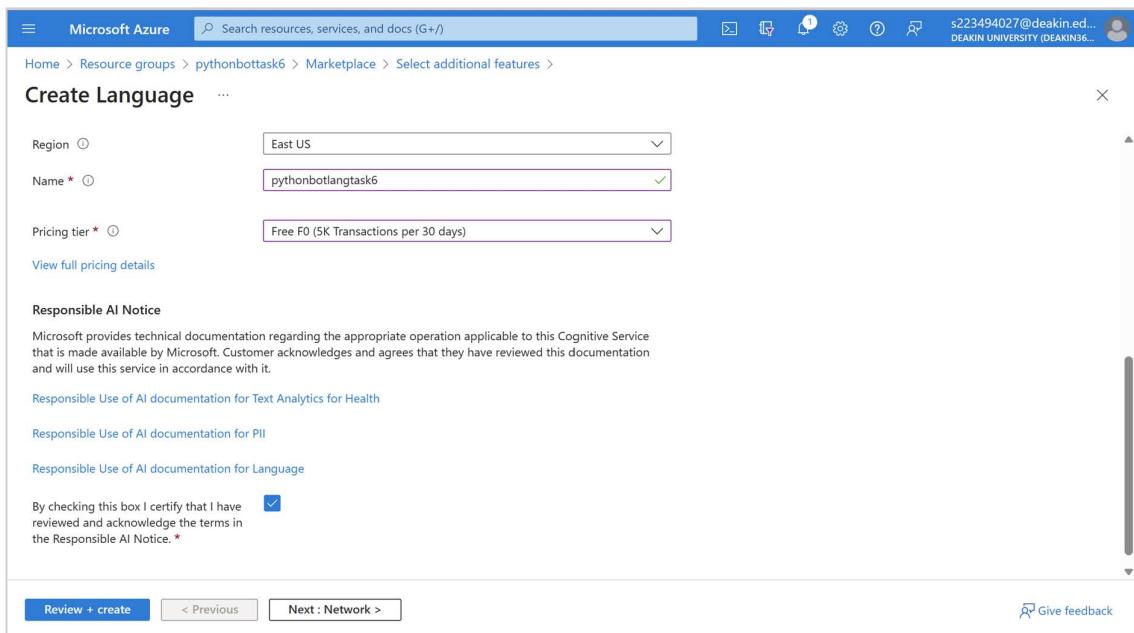


Fig 12 Create Language group

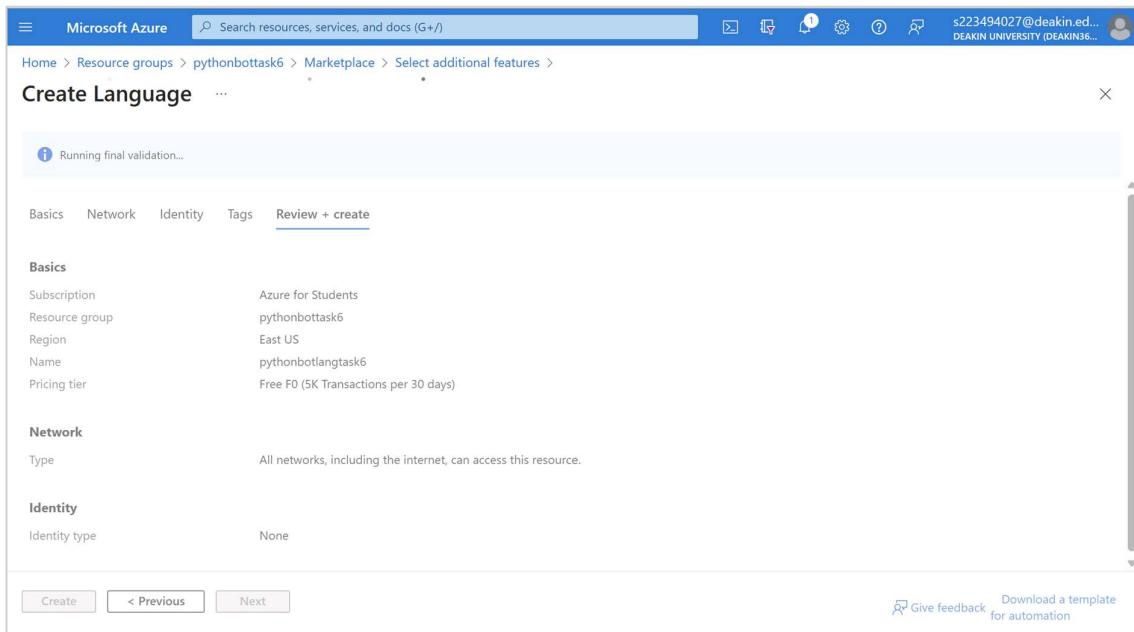


Fig 13: Validation process for creating language

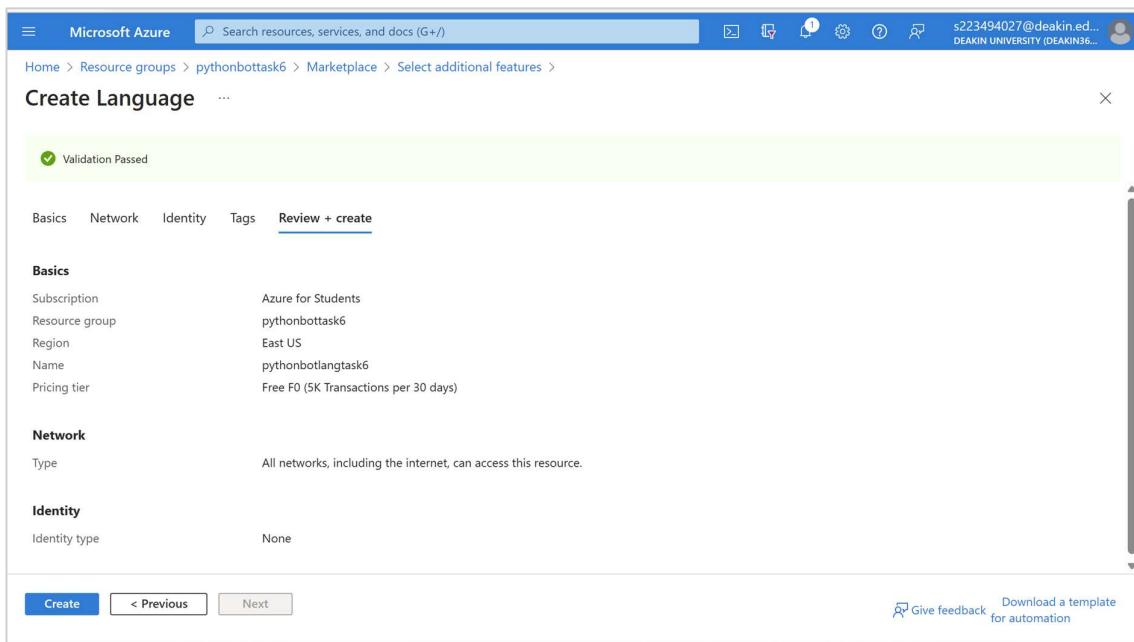


Fig 14 Validation process – Passed

- Once the validation is passed for language group. Then deployment of Text Analytics begins.

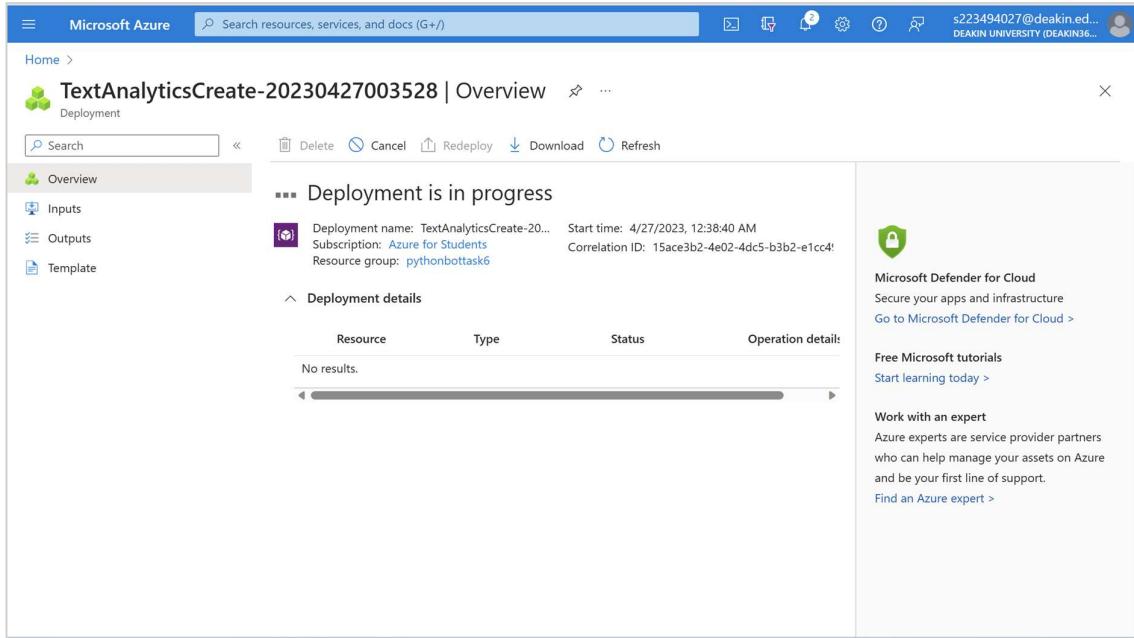


Fig 15 Deployment of language

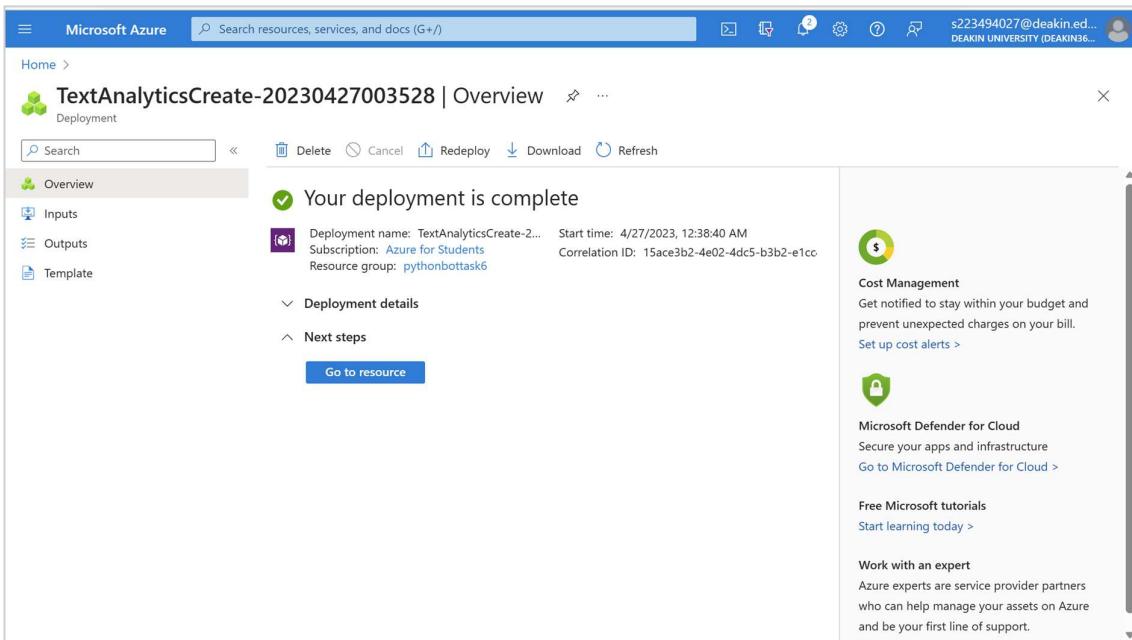


Fig 16 Deployment is completed

- Select Resources and select the language group that we have created. We will be getting the keys and endpoint will be available from overview page.

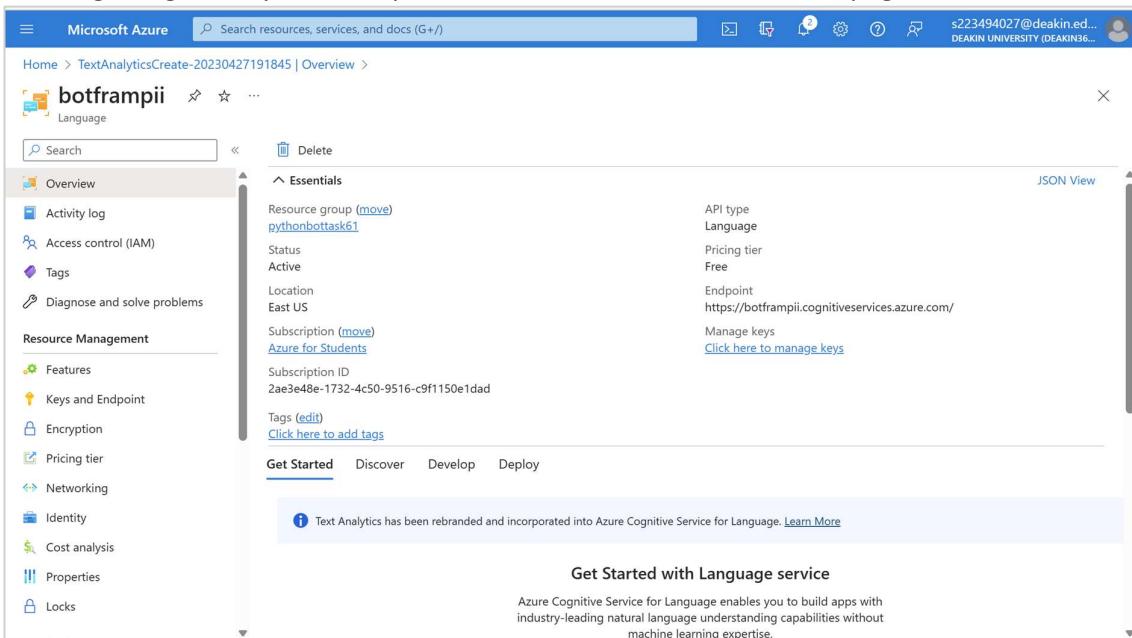


Fig 17: Language -overview

- Keep a note of the key and endpoints

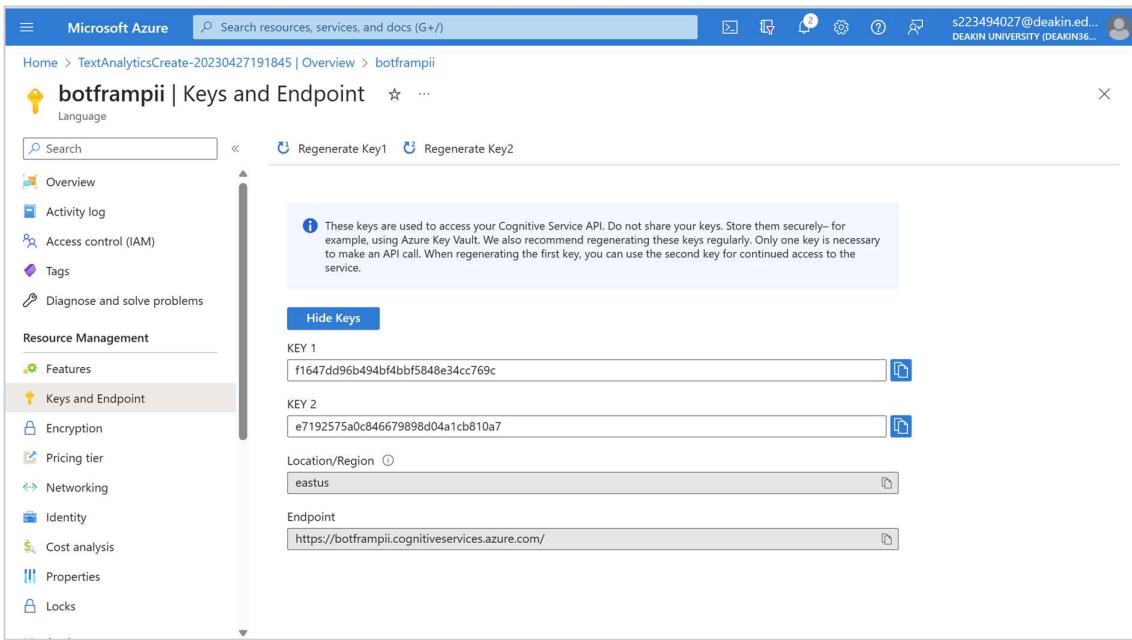


Fig 18 Keys & Endpoint

➤ Open Visual Studio by using <https://dev.botframework.com>

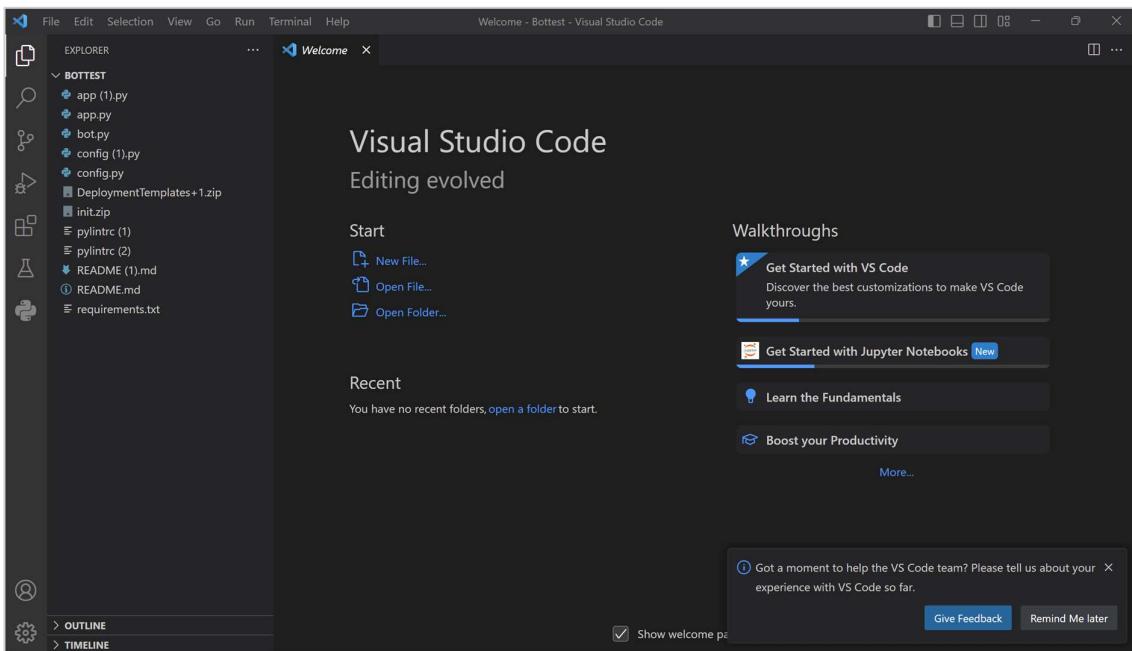


Fig 19 Visual Studio

- In Visual Studio / Windows PowerShell/ Command Prompt use any one of the ways to register Azure. Login to Azure by using the code “az login”

```
C:\Users\DeLL>az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.
```

*Fig 20 Azure login through prompt*

- We get a notification window through web browser. Required to login through web browser we will be getting all the details related to Azure subscription, ID, etc. We need to note down the ID for proceeding with the app registration and deployment.

```
"cloudName": "AzureCloud",
"homeTenantId": "d02378ec-1688-46d5-8540-1c28b5f470f6",
"id": "c094e16e-df8d-4a15-bbc5-0dd2487e788c",
"isDefault": false,
"managedByTenants": [],
"name": "Free Trial",
"state": "Disabled",
"tenantId": "d02378ec-1688-46d5-8540-1c28b5f470f6",
"user": {
  "name": "s223494027@deakin.edu.au",
  "type": "user"
},
{
  "cloudName": "AzureCloud",
"homeTenantId": "d02378ec-1688-46d5-8540-1c28b5f470f6",
"id": "2ae3e48e-1732-4c50-9516-c9f1150e1dad",
"isDefault": true,
"managedByTenants": [],
"name": "Azure for Students",
"state": "Enabled",
"tenantId": "d02378ec-1688-46d5-8540-1c28b5f470f6",
"user": {
  "name": "s223494027@deakin.edu.au",
  "type": "user"
}
]
C:\Users\DeLL>az account set --subscription "c094e16e-df8d-4a15-bbc5-0dd2487e788c"
```

*Fig 21 Azure credentials in detail*

- Display the resource group that we have created in Azure

```
C:\Users\DeLL>az ad app create --display-name "pythonbottask61"
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#applications/$entity",
  "addIns": [],
  "api": {
    "acceptMappedClaims": null,
    "knownClientApplications": [],
    "oauth2PermissionScopes": [],
    "preAuthorizedApplications": [],
    "requestedAccessTokenVersion": 2
  },
  "appId": "83c42cb4-a8a7-426c-bbd1-1686947b2dc3",
  "appRoles": [],
  "applicationTemplateId": null,
  "certification": null,
  "createdDateTime": "2023-04-27T15:09:39.0906235Z",
  "defaultRedirectUri": null,
  "deletedDateTime": null,
  "description": null,
  "disabledByMicrosoftStatus": null,
  "displayName": "pythonbottask61",
  "groupMembershipClaims": null,
  "id": "57b4c63c-9832-4aed-ac37-0d4fa3edaddc",
  "identifierUris": [],
  "info": {
    "logoUrl": null,
    "marketingUrl": null,
    "privacyStatementUrl": null,
    "supportUrl": null,
    "termsOfServiceUrl": null
  },
  "isDeviceOnlyAuthSupported": null,
```

*Fig 22 Resource group*

- Deploy the model by using the template and parameters. Here, we use the JSON files from the Deployment Template. First, we use “Deploy using existing resource group”. Here updates the resource group name, App service name, Region in the template file and save the file.

```

File Edit Selection View Go Run Terminal Help
parameters-for-template-BotApp-with-rg.json - Bottest - Visual Studio Code
EXPLORER
BOTTEST
  > .condo
  > .venv
  DeploymentTemplates
    > deployUseExistResourceGroup
      > parameters-for-template-AzureBot...
      > parameters-for-template-BotApp...
        > README.md
        > template-AzureBot-with-rg.json
        > template-BotApp-with-rg.json
      > deployWithNewResourceGroup
      > echo_bot
      & app (1).py
      & app.py
      & bot.py
      & config (1).py
      & config.py
      & init.py
      & init.zip
      & pylintc (1)
      & pylintc (2)
      & README (1).md
      & README.md
      & requirements.txt
1  {
2   "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentParameters.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "appServiceName": {
6       "value": "pythonbottest61"
7     },
8     "existingAppServicePlanName": {
9       "value": ""
10    },
11    "existingAppServicePlanLocation": {
12      "value": ""
13    },
14    "newAppServicePlanName": {
15      "value": "botframpii"
16    },
17    "newAppServicePlanLocation": {
18      "value": "East US"
19    },
20    "newAppServicePlanSku": {
21      "value": {
22        "name": "S1",
23        "tier": "Standard",
24        "size": "S1",
25        "family": "S",
26        "capacity": 1
27      }
28    }
29  }

```

Fig 23 Deployment Template for existing resources

- Use the code “az deployment group create –resource-group “<resource>” –template-file “<template.json>” –parameters “<@parameter.json>” . Take the file name from the deployment folder.

```

PS C:\Users\Del\OneDrive\Desktop\Bottest\DeploymentTemplates\deployUseExistResourceGroup> az deployment group create --resource-group "pythonbottask61" --template-file "template-BotApp-with-rg.json" --parameters "@parameters-for-template-BotApp-with-rg.json"
{
  "id": "/subscriptions/2ae3e48e-1732-4c50-9516-c9f1150e1dad/resourceGroups/pythonbottask61/providers/Microsoft.Resources/deploymentGroups/template-BotApp-with-rg",
  "location": null,
  "name": "template-BotApp-with-rg",
  "properties": {
    "correlationId": "c956bff3-2cb8-434c-b7fe-f9151b016985",
    "debugSetting": null,
    "dependencies": [
      {
        "dependsOn": [
          {
            "id": "/subscriptions/2ae3e48e-1732-4c50-9516-c9f1150e1dad/resourceGroups/pythonbottask61/providers/Microsoft.Web/serverFarms/botframpii",
            "resourceGroup": "pythonbottask61",
            "resourceName": "botframpii",
            "resourceType": "Microsoft.Web/serverFarms"
          }
        ],
        "id": "/subscriptions/2ae3e48e-1732-4c50-9516-c9f1150e1dad/resourceGroups/pythonbottask61/providers/Microsoft.Web/sites/pythonbottest61",
        "resourceGroup": "pythonbottask61",
        "resourceName": "pythonbottest61",
        "resourceType": "Microsoft.Web/sites"
      },
      {
        "dependsOn": [
          {
            "id": "/subscriptions/2ae3e48e-1732-4c50-9516-c9f1150e1dad/resourceGroups/pythonbottask61/providers/Microsoft.Web/sites/pythonbottest61",
            "resourceGroup": "pythonbottask61"
          }
        ]
      }
    ]
  }
}

```

Fig 24 Deployment

- After the successful execution, next we will deploy the Azure bot template. We will update the Azure Bot ID from Azure portal. Update the same.

```

1  {
2      "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentParameters.json#",
3      "contentVersion": "1.0.0.0",
4      "parameters": {
5          "azureBotId": {
6              "value": "pythontask006"
7          },
8          "azureBotSku": {
9              "value": "S1"
10         },
11         "azureBotRegion": {
12             "value": "global"
13         },
14         "botEndpoint": {
15             "value": ""
16         },
17         "appId": {
18             "value": ""
19         }
20     }
21 }

```

Fig 25 Deploy Azure Bot template

- If we check the resource, we will be able to view App Service is been listed under the resource group.

Name	Type	Location	...
botframpii	App Service plan	East US	...
botframpii	Storage account	East US	...
pythonbottask	Storage account	East US	...
pythonbottest61	App Service	East US	...

Fig 26 Resource group

- For the next codes to deployment, we need to note down the default domain.

**pythonbottest61** Overview

**Essentials**

- Resource group: pythonbottest61
- Status: Running
- Location: East US
- Subscription: Azure for Students
- Subscription ID: 2ae3e48e-1732-4c50-9516-c9f1150e1dad
- Tags: Click here to add tags

**Properties** Monitoring Logs Capabilities Notifications Recommendations

**Web app**

- Name: pythonbottest61
- Publishing model: Code
- Runtime Stack: Python - 3.7

**Domains**

- Default domain: pythonbottest61.azurewebsites.net

Fig 27 App Services

- Update the default domain as endpoint and update the Azure ID for deploying the Azure Bot App. Save the files

```

parameters-for-template-AzureBot-with-rg.json - Bottest - Visual Studio Code
config.py
parameters-for-template-BotApp-with-rg.json
parameters-for-template-AzureBot-with-rg.json
DeploymentTemp

DeploymentTemplates > deployUseExistResourceGroup > parameters-for-template-AzureBot-with-rg.json > {} parameters > {} botEndpoint > value
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentParameters.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "azureBotId": {
            "value": "pythontask006"
        },
        "azureBotSku": {
            "value": "S1"
        },
        "azureBotRegion": {
            "value": "global"
        },
        "botEndpoint": {
            "value": "https://pythonbottest61.azurewebsites.net/api/messages"
        },
        "appId": {
            "value": "2ae3e48e-1732-4c50-9516-c9f1150e1dad"
        }
    }
}

```

Fig 28 Template for Azure Bot

➤ Update the endpoint and Azure ID in Bot.py file

```

# Copyright (c) Microsoft Corporation. All rights reserved.
# Licensed under the MIT License.

from botbuilder.core import ActivityHandler, TurnContext
from botbuilder.schema import ChannelAccount

import os
from azure.core.credentials import AzureKeyCredential
from azure.ai.textanalytics.aio import TextAnalyticsClient

# Language Resource credentials
endpoint = "https://botframpi.cognitiveservices.azure.com/"
key = "f1647dd96d494bfabbf848e34cc769c"

class MyBot(ActivityHandler):
    async def on_message_activity(self, turn_context: TurnContext) -> None:
        documents = []

        text_analytics_client = TextAnalyticsClient(
            endpoint=endpoint, credential=AzureKeyCredential(key)
        )

        documents.append(turn_context.activity.text)

        async with text_analytics_client:
            result = await text_analytics_client.recognize_pii_entities(documents)

```

Fig 29 Bot.py

➤ Endpoint is selected from Azure from the bot app service that we have recently created. The end point is updated from “Default domain”

Name	Type	Location	...
botframpi	Language	East US	...
botframpi	App Service plan	East US	...
pythonbottask	Storage account	East US	...
pythonbottest61	App Service	East US	...
pythontask006	Azure Bot	Global	...

Fig 30 Resource Group > App Services

The screenshot shows the Microsoft Azure portal interface for a bot named "pythonbottest61". The top navigation bar includes the Microsoft Azure logo, a search bar, and user information (s223494027@deakin.edu.. DEAKIN UNIVERSITY (DEAKIN36..)). Below the navigation is a breadcrumb trail: Home > TextAnalyticsCreate-20230427191845 | Overview > botframpii > pythonbottest61 >. The main content area displays the bot's details under the "Web App" category. The "Overview" tab is selected, showing the following key information:

- Resource group:** pythonbottest61
- Status:** Running
- Location:** East US
- Subscription:** Azure for Students
- Subscription ID:** 2ae3e48e-1732-4c50-9516-c9f1150e1dad
- Tags:** Click here to add tags
- Properties:** Web app (Name: pythonbottest61, Publishing model: Code, Runtime Stack: Python - 3.7), Domains (Default domain: pythonbottest61.azurewebsites.net)

On the left sidebar, other tabs like Activity log, Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment slots, Deployment Center, Configuration, Authentication, Application Insights, and Identity are visible.

Fig 31 Bot overview

## ➤ Keys and Endpoints are collected from language created

The screenshot shows the Microsoft Azure portal interface for the "Language" service of the bot "botframpii". The top navigation bar and breadcrumb trail are identical to Fig 31. The main content area displays the language service's details under the "Language" category. The "Overview" tab is selected, showing the following key information:

- Resource group:** pythonbottest61
- Status:** Active
- Location:** East US
- Subscription:** Azure for Students
- Subscription ID:** 2ae3e48e-1732-4c50-9516-c9f1150e1dad
- Tags:** Click here to add tags
- API type:** Language
- Pricing tier:** Free
- Endpoint:** https://botframpii.cognitiveservices.azure.com/
- Manage keys:** Click here to manage keys

On the left sidebar, other tabs like Features, Keys and Endpoint (selected), Encryption, Pricing tier, Networking, Identity, Cost analysis, Properties, and Locks are visible. A note at the bottom states: "Text Analytics has been rebranded and incorporated into Azure Cognitive Service for Language. [Learn More](#)".

Fig 32 Overview Language

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes 'Microsoft Azure', a search bar, and user information ('s223494027@deakin.edu.. DEAKIN UNIVERSITY (DEAKIN36..)'). Below the navigation is a breadcrumb trail: Home > TextAnalyticsCreate-20230427191845 | Overview > botframpii > pythonbottask61 > pythonbottest61 > pythonbottask61 > botframpii. The main content area is titled 'botframpii | Keys and Endpoint'. On the left, there's a sidebar with 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Resource Management' (with 'Features', 'Keys and Endpoint' selected), 'Encryption', 'Pricing tier', 'Networking', 'Identity', 'Cost analysis', 'Properties', and 'Locks'. The 'Keys and Endpoint' section contains two key fields: 'KEY 1' with value 'f1647dd96b494bf4bbf5848e34cc769c' and 'KEY 2' with value 'e7192575a0c846679898d04a1cb810a7'. It also shows 'Location/Region' as 'eastus' and 'Endpoint' as 'https://botframpii.cognitiveservices.azure.com/'. A note at the top says: 'These keys are used to access your Cognitive Service API. Do not share your keys. Store them securely—for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.' There are 'Regenerate Key1', 'Regenerate Key2', and 'Hide Keys' buttons.

Fig 33 Key and Endpoint

The screenshot shows the Visual Studio Code interface with the 'Bot test' project open. The Explorer sidebar shows files like '.conda', '.venv', 'DeploymentTemplates', 'deployUseExistResourceGroup', 'parameters-for-template-AzureBo...', 'parameters-for-template-BotApp...', 'readme.md', 'template-AzureBot-with-rg.json', 'template-BotApp-with-rg.json', 'echo\_bot', 'app (1).py', 'app.py', and 'bot.py'. The 'bot.py' file is selected and shown in the editor pane. The code in 'bot.py' is as follows:

```

1 # Copyright (c) Microsoft Corporation. All rights reserved.
2 # Licensed under the MIT License.
3
4 from botbuilder.core import ActivityHandler, TurnContext
5 from botbuilder.schema import ChannelAccount
6
7 import os
8 from azure.core.credentials import AzureKeyCredential
9 from azure.ai.textanalytics.aio import TextAnalyticsClient
10
11 # Language Resource credentials
12 endpoint = "https://botframpii.cognitiveservices.azure.com/"
13 key = "f1647dd96b494bf4bbf5848e34cc769c"
14
15 class MyBot(ActivityHandler):
16     async def on_message_activity(self, turn_context: TurnContext) -> None:
17         documents = []
18
19         text_analytics_client = TextAnalyticsClient(
20             endpoint=endpoint, credential=AzureKeyCredential(key)
21         )
22
23         documents.append(turn_context.activity.text)
24
25         async with text_analytics_client:
26             result = await text_analytics_client.recognize_pii_entities(documents)
    
```

Fig 34 Bot py updated

- Once the file is updated, we have to zip all the requirement, deployment template, app.py, bot.py, config, init, pylintc and readme files to a single zip file under the main drive folder. Use the below codes for publishing the bot to Azure.

```

PS C:\Users\DeLL\OneDrive\Desktop\Bottest> az webapp deployment source config-zip --resource-group "pythonbottask61" --name "pythonbottest61" --src "Botapp.zip"
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
    
```

Fig 35 Publishing the bot to Azure

```

PS C:\Users\DELL\OneDrive\Desktop\Bottest> az webapp deployment source config-zip --resource-group "pythonbottask61" --name "pythonbottest61" --src "Botapp.zip"
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
{
    "active": true,
    "author": "N/A",
    "author_email": "N/A",
    "build_summary": {
        "errors": [],
        "warnings": []
    },
    "complete": true,
    "deployer": "Push-Deployer",
    "end_time": "2023-04-27T18:41:00.7405616Z",
    "id": "cb3dc8fb-82ed-4050-88c0-5331e8698d2b",
    "is_READONLY": true,
    "is_temp": false,
    "last_success_end_time": "2023-04-27T18:41:00.7405616Z",
    "log_url": "https://pythonbottest61.scm.azurewebsites.net/api/deployments/latest/log",
    "message": "Created via a push deployment",
    "progress": "",
    "received_time": "2023-04-27T18:38:54.5875067Z",
    "site_name": "pythonbottest61",
    "start_time": "2023-04-27T18:38:56.0132422Z",
    "status": 4,
    "status_text": "",
    "url": "https://pythonbottest61.scm.azurewebsites.net/api/deployments/latest"
}
PS C:\Users\DELL\OneDrive\Desktop\Bottest>

```

Fig 36 Bot successfully published

- Open the Azure portal, we will be able to view the Azure bot registered under the resource group.

Name	Type	Location
botframpii	Language	East US
botframpii	App Service plan	East US
pythonbottask	Storage account	East US
pythonbottest61	App Service	East US
pythontask006	Azure Bot	Global

Fig 37 Resource group Overview

➤ Open the Azure Bot registered under the resource group

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below the navigation bar, the URL path shows the bot's location: ... > pythonbottest61 > pythonbottask61 > botframpii > pythonbottask61 > pythonbottest61 > pythonbottask61 > botframpii > pythonbottask61 > pythontask006. On the right side of the header, there are user details: s223494027@deakin.edu.. DEAKIN UNIVERSITY (DEAKIN36...). The main content area is titled 'pythontask006' and 'Azure Bot'. It features a left sidebar with navigation links: Overview, Activity log, Access control (IAM), Tags, Settings (which is currently selected), Bot profile, Configuration, Channels, Pricing, Test in Web Chat, Encryption, Networking, Properties, Locks, and Monitoring. The main panel shows 'Help us improve Bot Service. Take our survey!' and a section titled 'Essentials' with details: Resource group (move) pythonbottask61, Subscription (move) Azure for Students, Subscription ID 2ae3e48e-1732-4c50-9516-c9f1150e1dad, and Tags (edit) Click here to add tags. A 'JSON View' link is also present. Below this, there's a promotional section for 'Build enterprise-grade conversational AI'.

Fig 38 Azure Bot overview

➤ Test the bot using Test in Wb Chat

This screenshot shows the 'Test in Web Chat' interface for the 'pythontask006' bot. The left sidebar is identical to Fig 38, with 'Test in Web Chat' selected. The main panel has a title 'pythontask006 | Test in Web Chat'. It contains a 'Test' section with a 'Start over' button and a message input field labeled 'Type your message'.

Fig 39 Test the Bot in Web chat

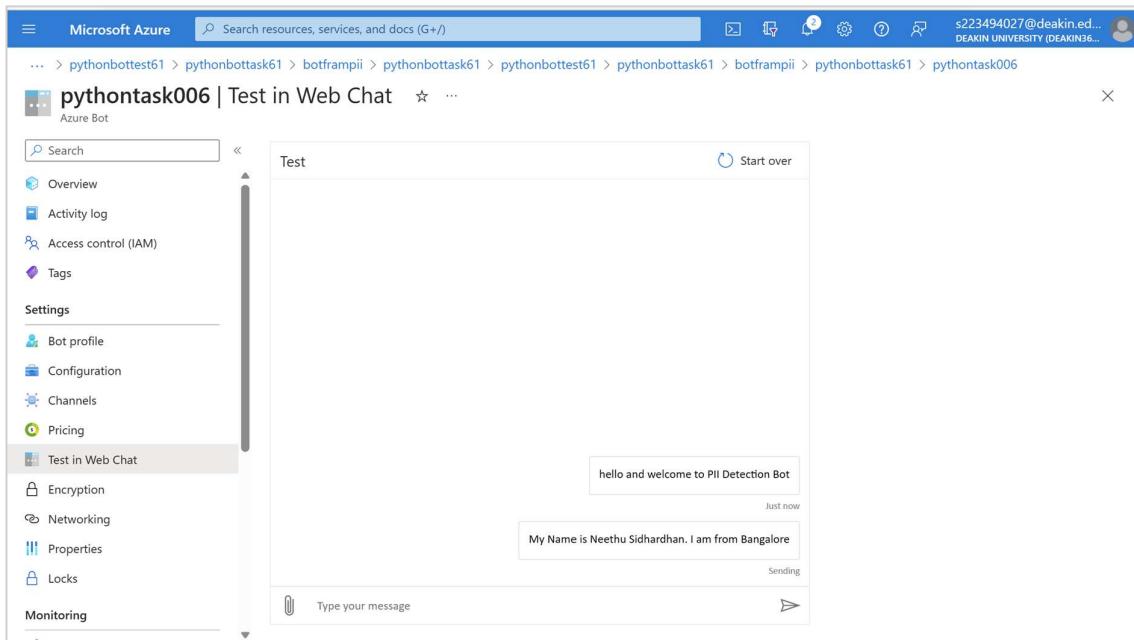


Fig 40 Testing the Bot

## ➤ Create a bot with Bot Framework SDK

A screenshot of the Microsoft Learn website at https://learn.microsoft.com/en-au/training/modules/design-bot-conversation-flow/6-exercise-create-a-bot. The top navigation bar shows 'Training' (selected), 'Products', 'Career Paths', 'Learning paths', 'Courses', 'Educator Center', 'Student hub', 'FAQ &amp; Help'. Below that is a breadcrumb trail: 'Learn / Training / Browse / Create conversational AI solutions / Create a bot with the Bot Framework SDK /'. The main content area is titled 'Exercise - Create a bot with the Bot Framework SDK' (Unit 6 of 8, 100 XP). It says '30 minutes'. A note says 'This unit includes a lab to complete. Use the free resources provided in the lab to complete the exercises in this unit. You will not be charged.' Another note says 'Microsoft provides this lab experience and related content for educational purposes. All presented information is owned by Microsoft and intended solely for learning about the covered products and services in this Microsoft Learn module.' A 'Sign in to launch the lab' button is shown. A purple note at the bottom says 'To complete this exercise, you will need a Microsoft Azure subscription. If you don't already have one, you can sign up for a free trial at https://azure.com/free.'.

Fig 41 Lab for framework SDK

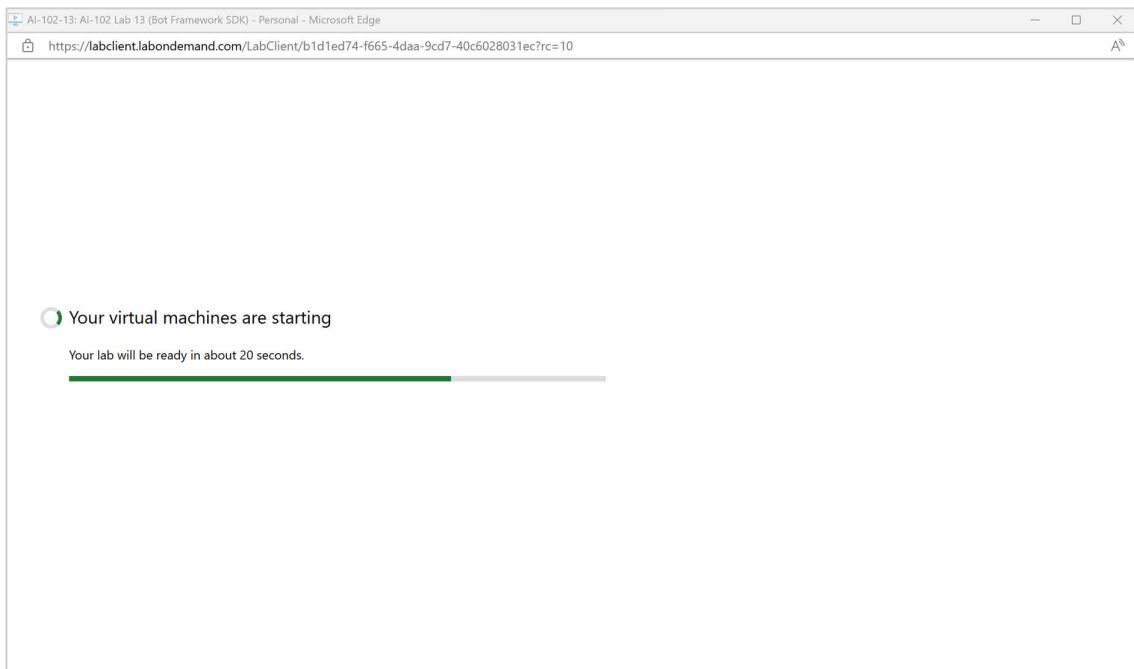


Fig 42 Virtual Machine loading

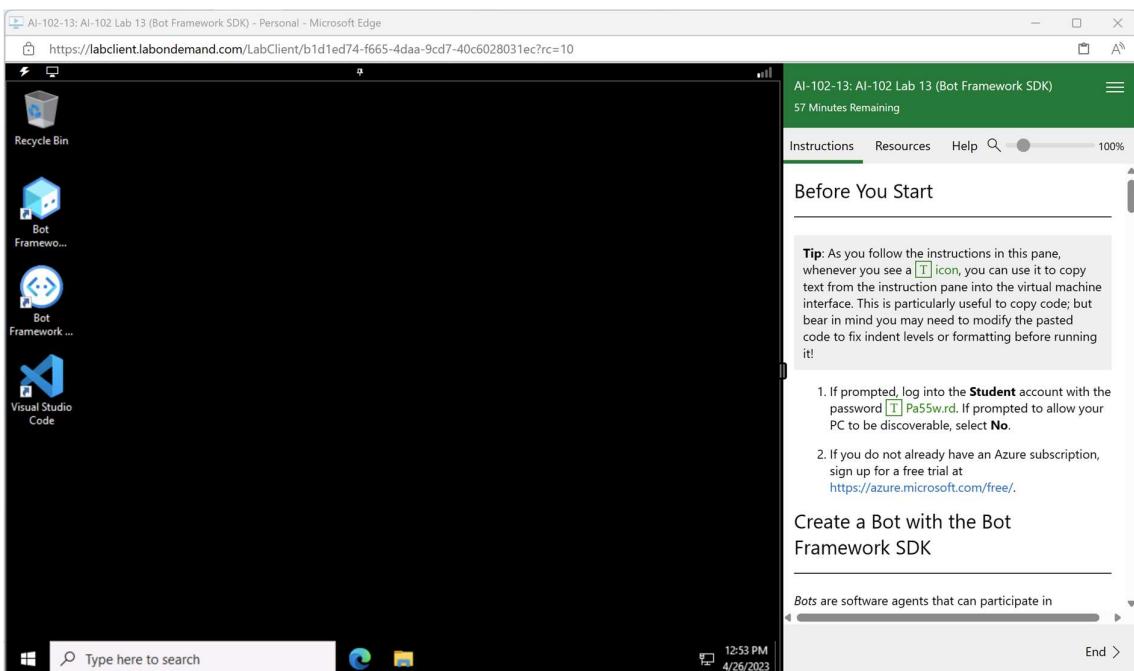
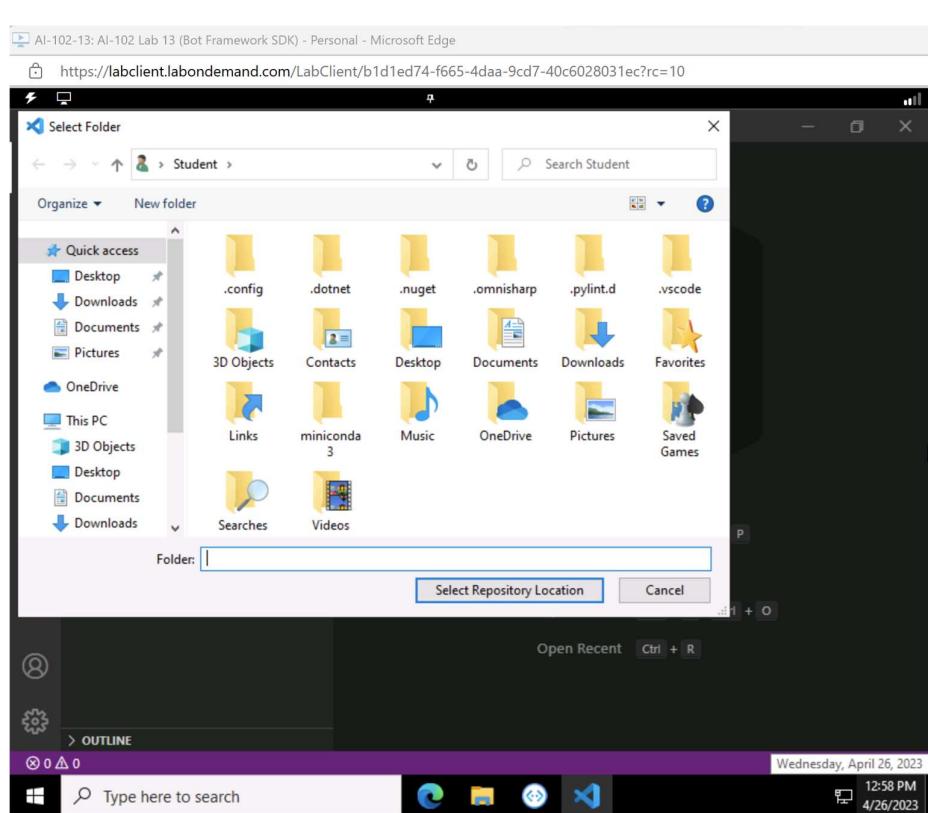
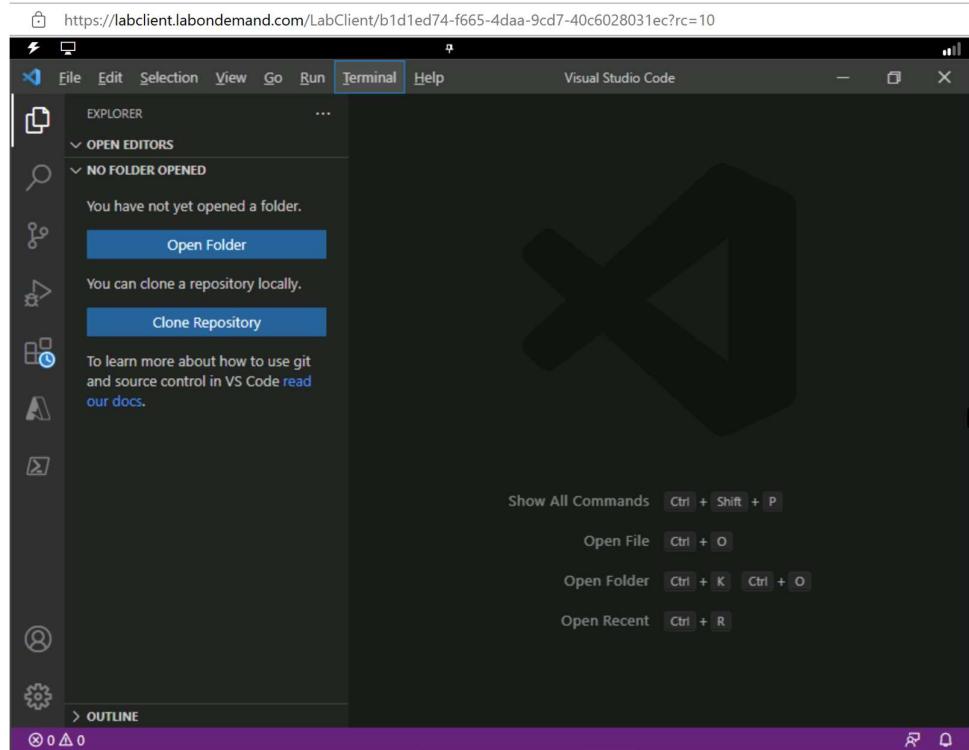


Fig 43 Virtual Machine Desktop

➤ Open the Terminal



➤ Install the requirement

```
pip install botbuilder-core  
pip install asyncio  
pip install aiohttp  
pip install cookiecutter==1.7.0
```

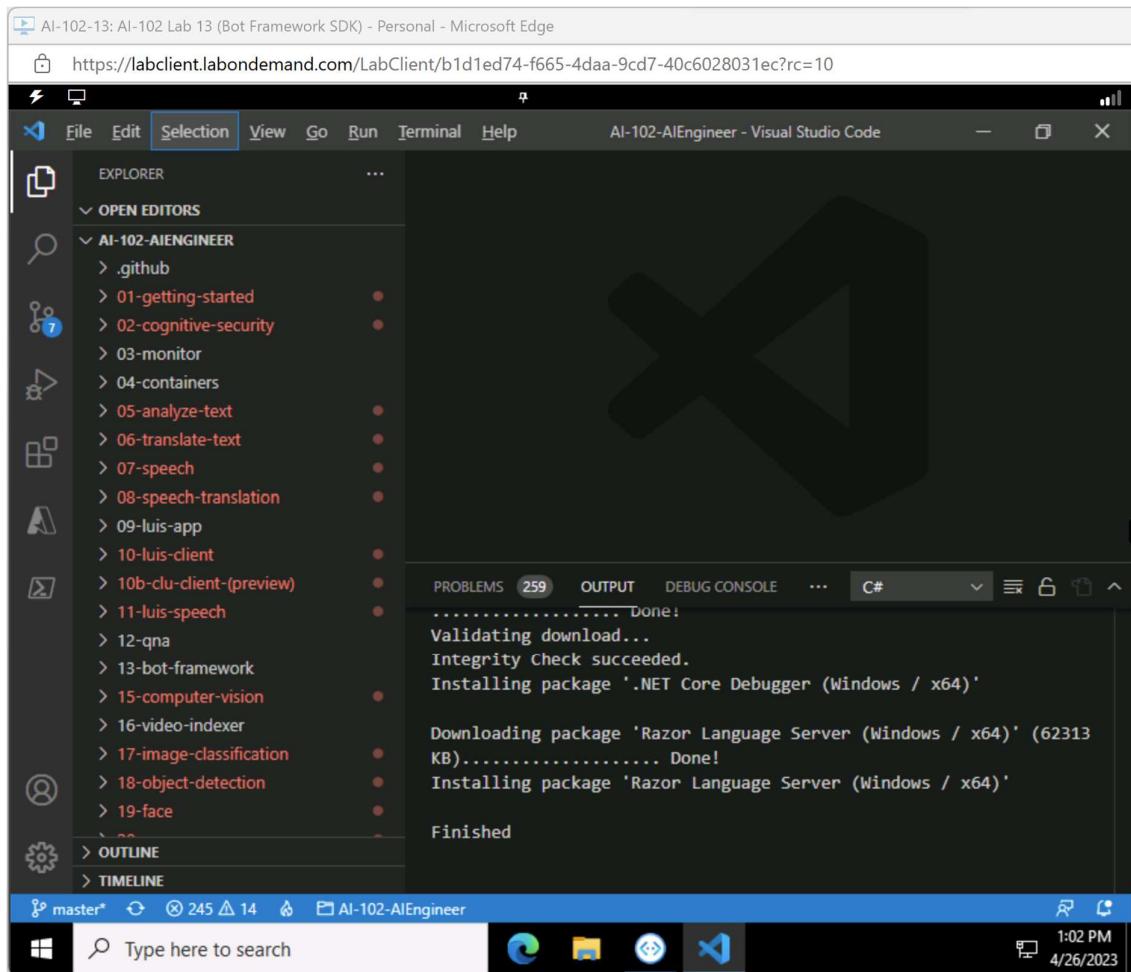
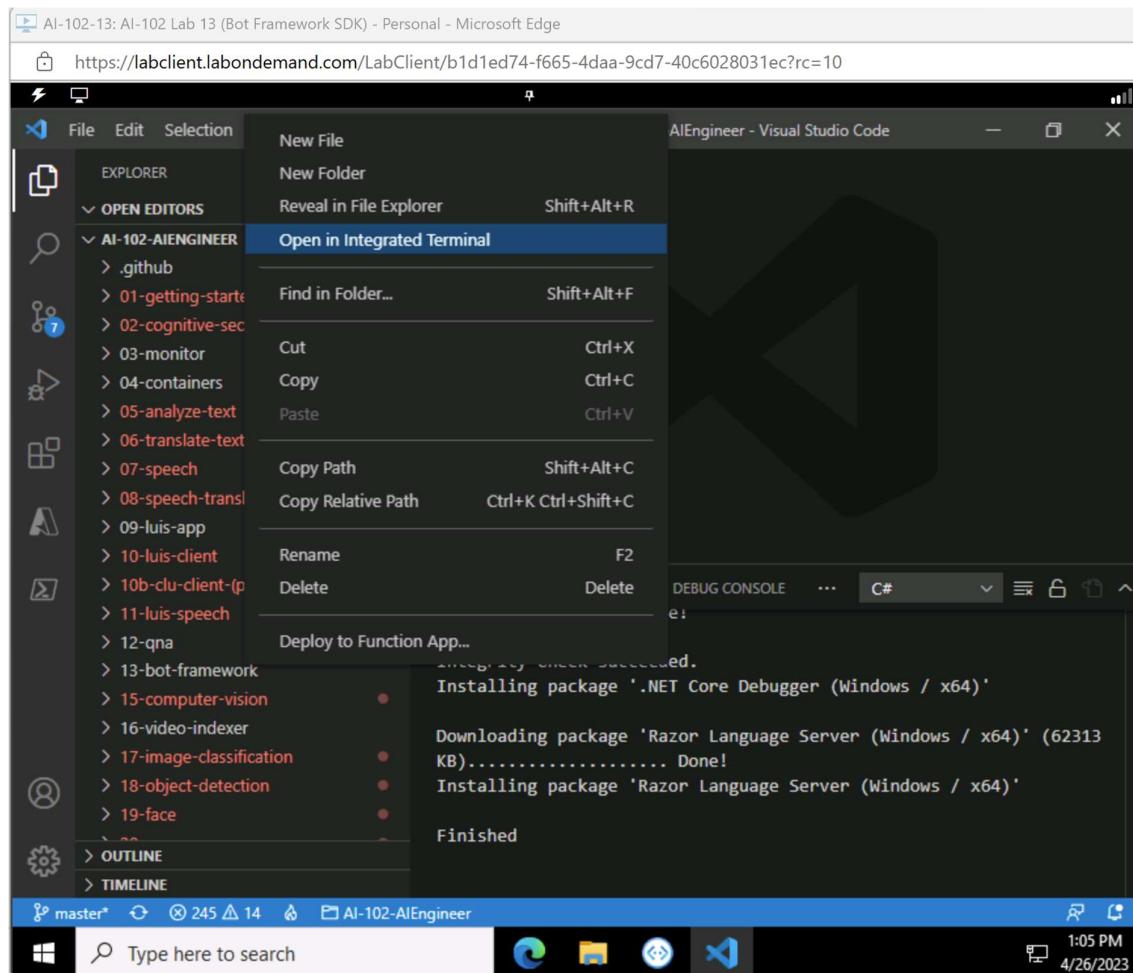


Fig 46 Install the packages



## Python

```
Python
[T] pip install botbuilder-core
      pip install asyncio
      pip install aiohttp
      pip install cookiecutter==1.7.0
```

4. After the templates and packages have been installed, run the following command to create a bot based on the *EchoBot* template:

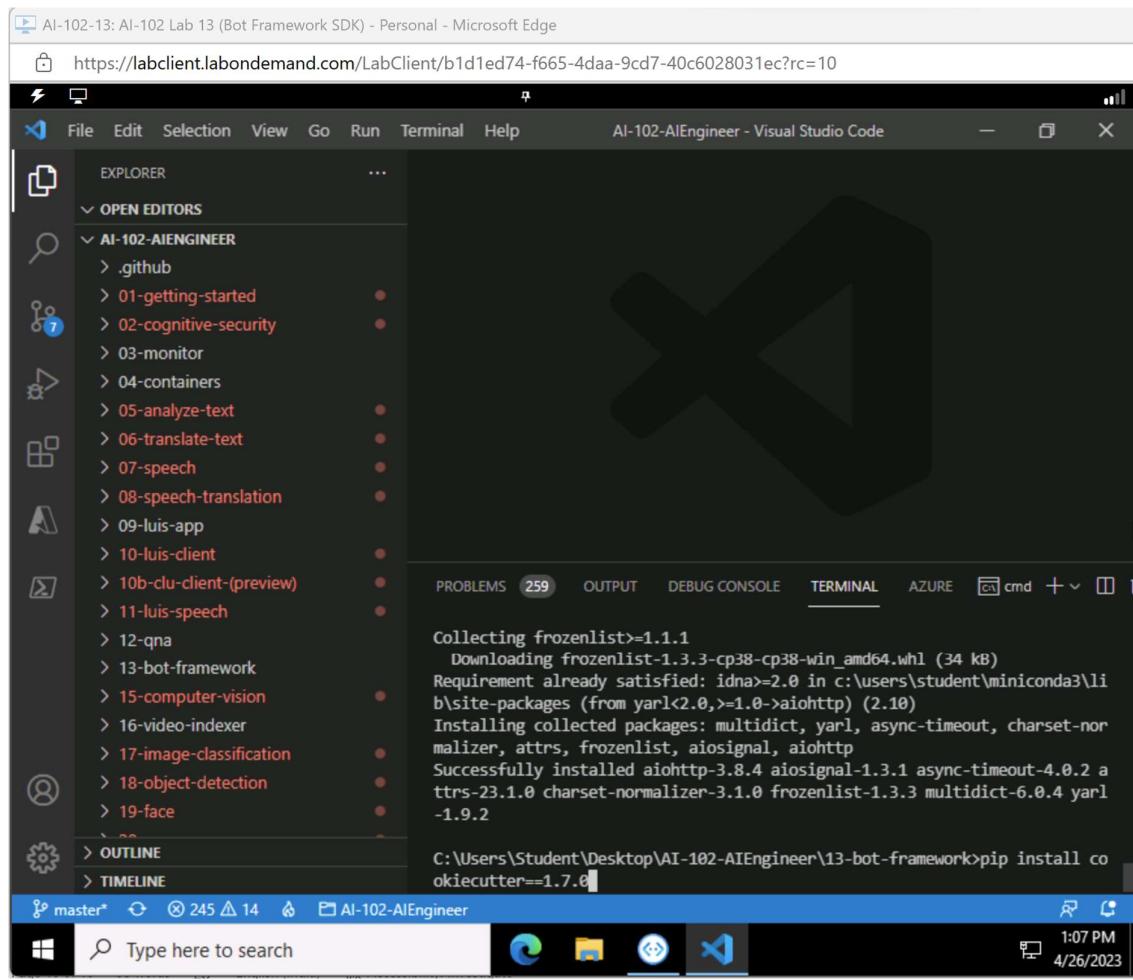


Fig 47 Install Cookie cutter

cookiecutter <https://github.com/microsoft/botbuilder-python/releases/download/Templates/echo.zip>

➤ Select the folder 13 for Bot framework

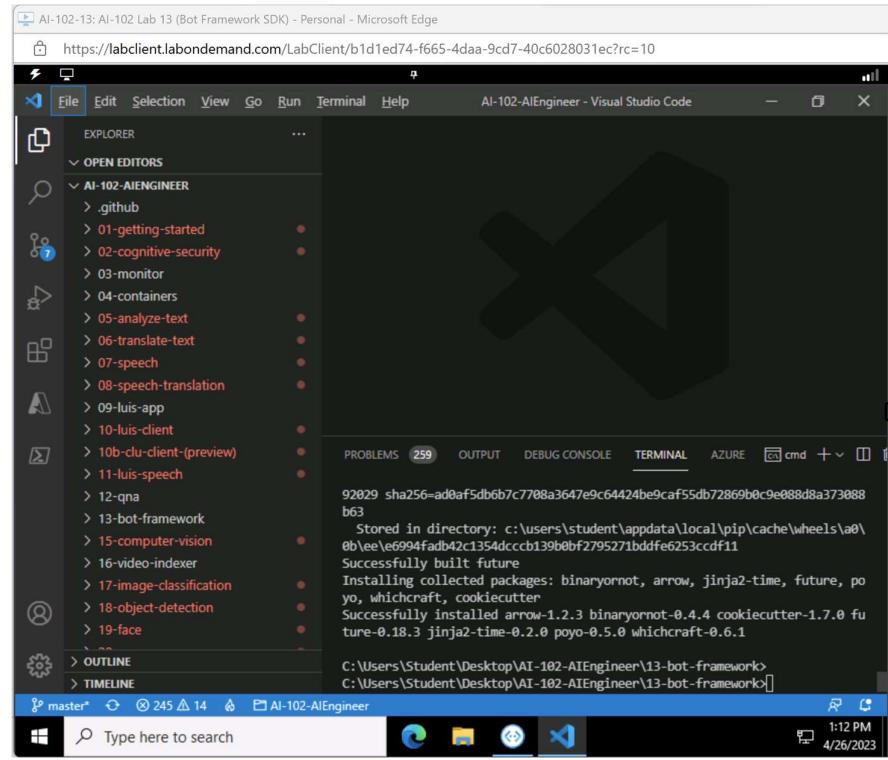


Fig 48 bot framework

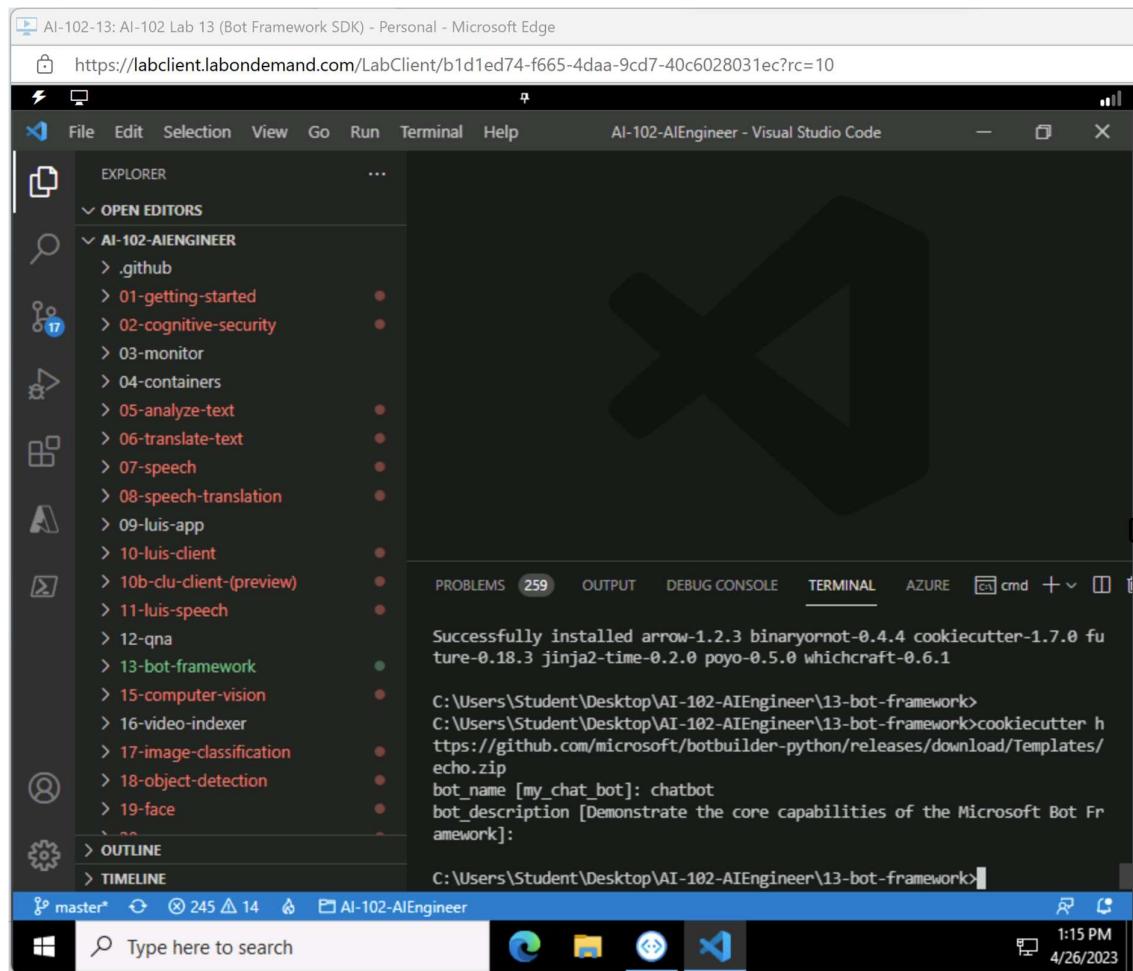
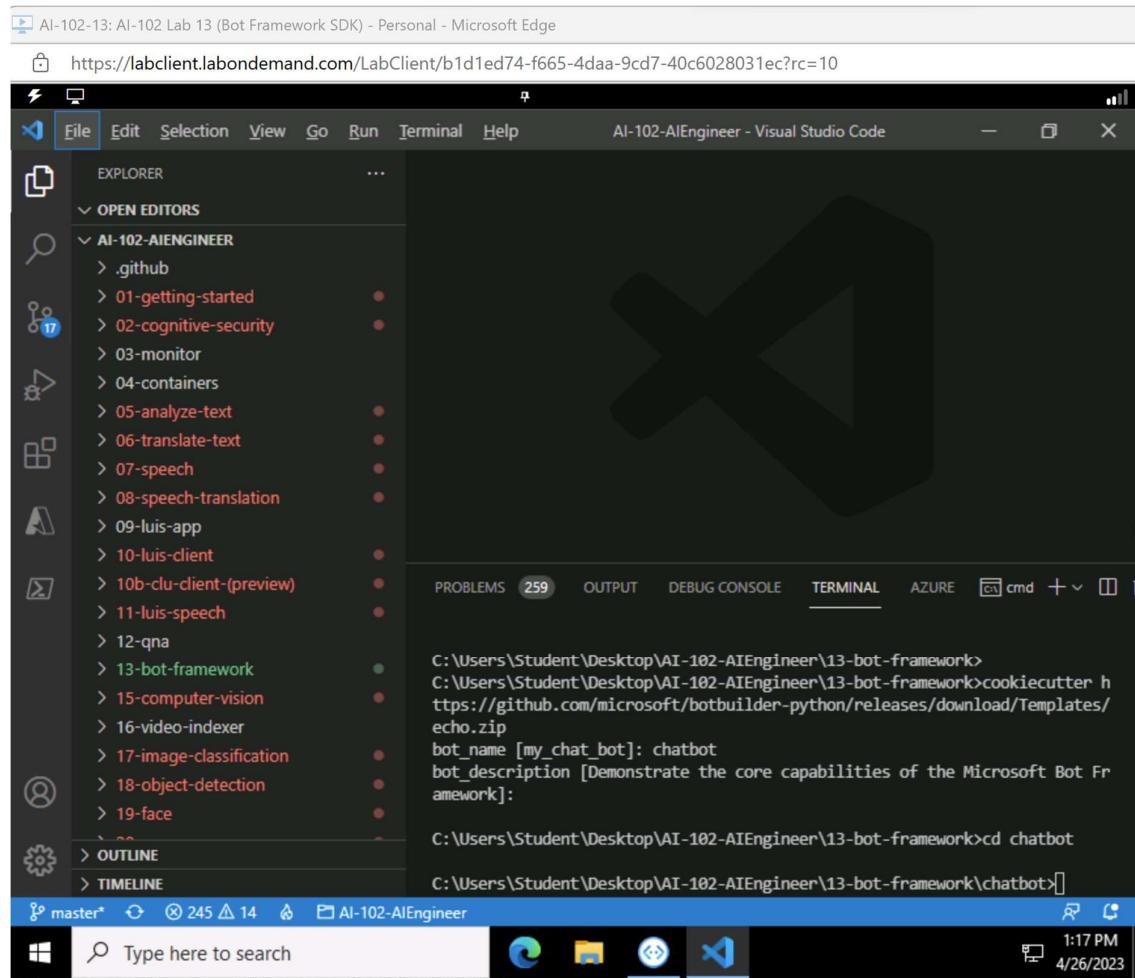


Fig 49 Give the Name for the bot name and Description



```
Python
python app.py
```

'hen the bot starts, note the endpoint at which it is

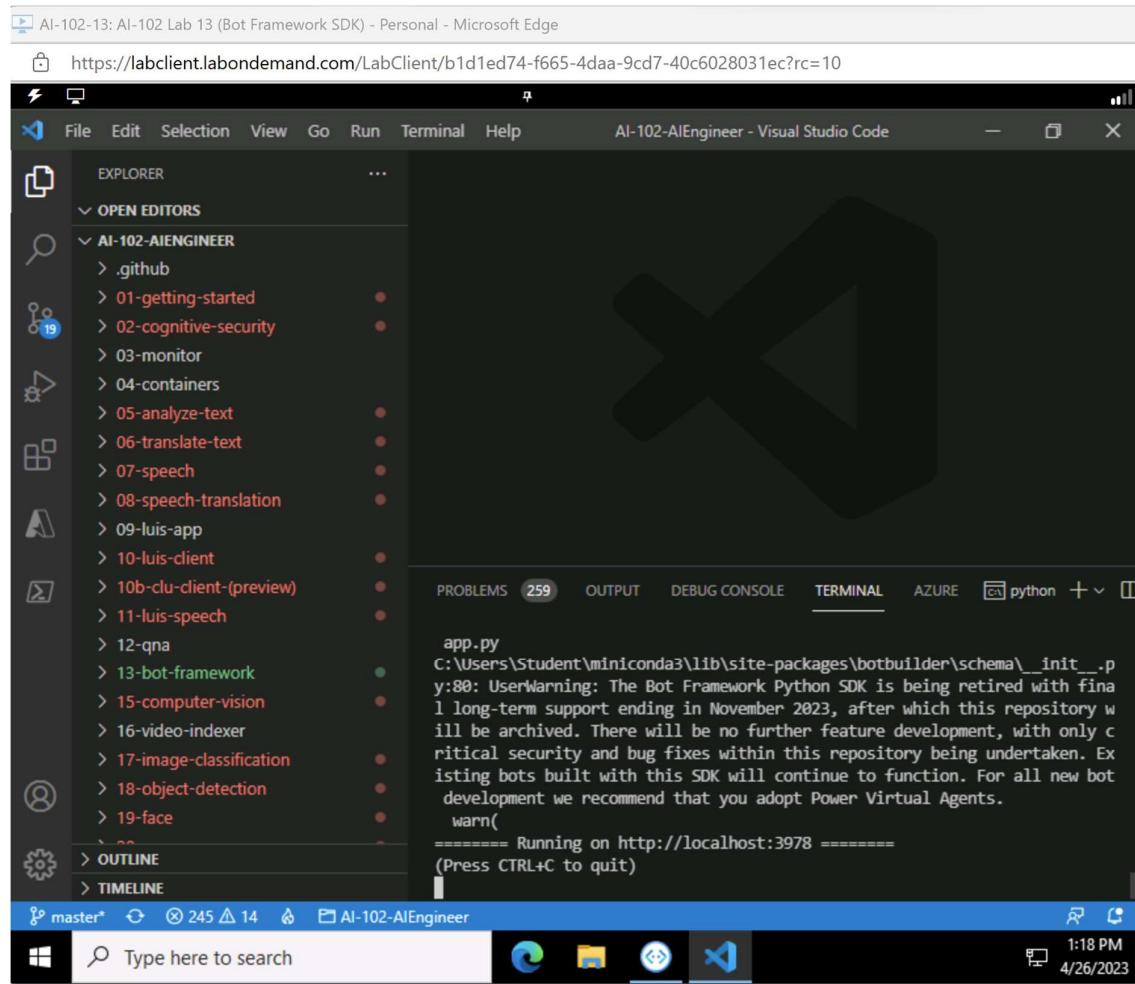


Fig 50 Hots name

The host's name is used for running the bot in Bot Emulator

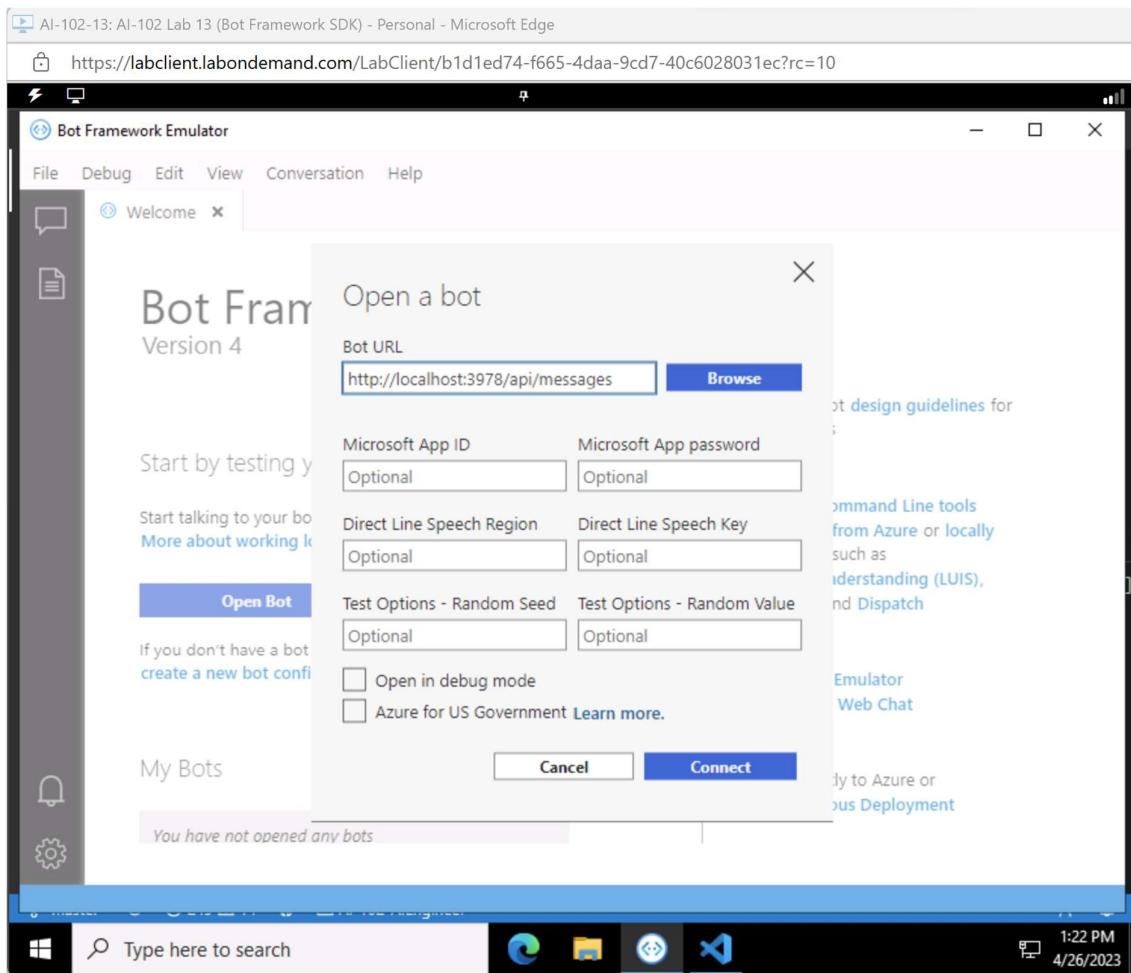


Fig 51 Bot Emulator with host name

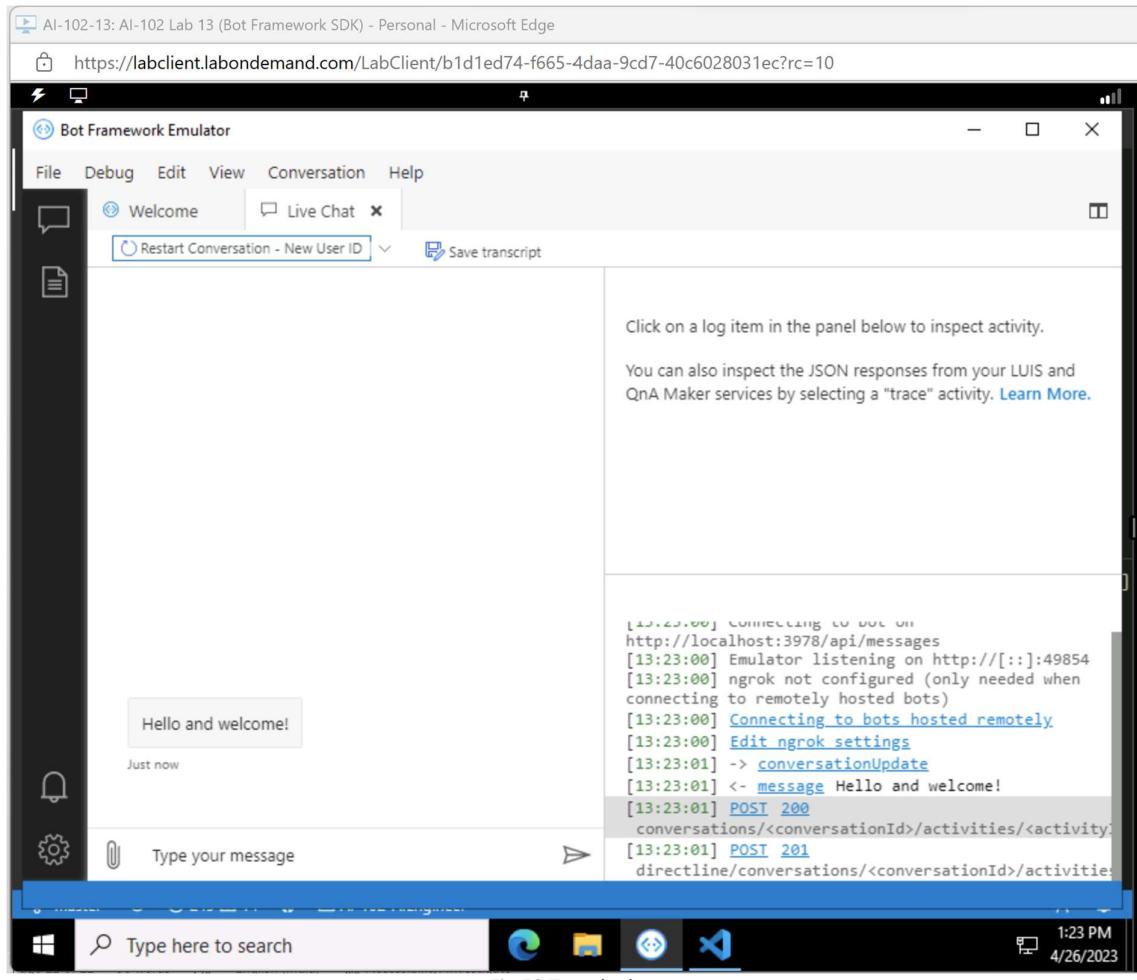


Fig 52 Test the bot

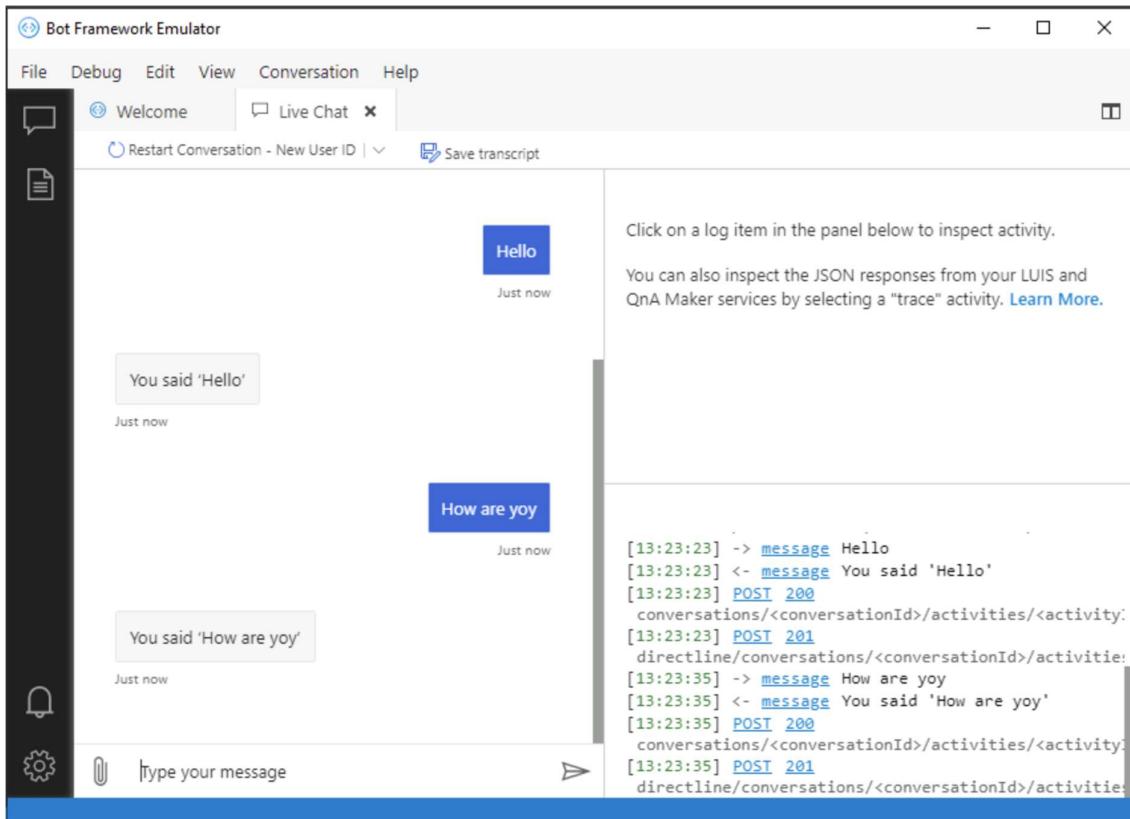


Fig 53 Echo Bot

## References

1. Haley, B., 2022. *GitHub*. [Online]  
Available at: <https://github.com/microsoft/BotBuilder-Samples/tree/main/samples/python/13.core-bot>  
[Accessed 27 April 2023].
2. Joseph, n.d. *Great Learning \_Azure bot walk through*. [Online]  
Available at:  
[https://olympus.mygreatlearning.com/courses/91262/pages/azurebot1?module\\_item\\_id=3987189](https://olympus.mygreatlearning.com/courses/91262/pages/azurebot1?module_item_id=3987189)
3. Microsoft, n.d. *Create a bot with the Bot Framework SDK*. [Online]  
Available at: <https://learn.microsoft.com/en-gb/azure/bot-service/bot-service-quickstart-create-bot?view=azure-bot-service-4.0&tabs=python%2Cvs>  
[Accessed 27 April 2023].
4. Microsoft, n.d. *Create a bot with the Bot Framework SDK*. [Online]  
Available at: <https://learn.microsoft.com/en-gb/azure/bot-service/bot-service-quickstart-create-bot?view=azure-bot-service-4.0&tabs=python%2Cvs>
5. Microsoft, n.d. *Debug an SDK-first bot / Microsoft*. [Online]  
Available at: <https://learn.microsoft.com/en-us/azure/bot-service/bot-service-debug-bot?source=recommendations&view=azure-bot-service-4.0&tabs=python>
6. Microsoft, n.d. *https://learn.microsoft.com/en-gb/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0*. [Online]  
Available at: <https://learn.microsoft.com/en-gb/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0>

- [overview?view=azure-bot-service-4.0](#)  
[Accessed 27 April 2023].
7. Microsoft, n.d. *Use the Azure portal to Create an Azure Bot resource.* [Online]  
Available at: <https://learn.microsoft.com/en-gb/azure/bot-service/abs-quickstart?view=azure-bot-service-4.0&tabs=userassigned>  
[Accessed April 2023].
  8. Sharma, C., n.d. *GitHub.* [Online]  
Available at: <https://github.com/microsoft/BotBuilder-Samples>  
[Accessed 27 April 2023].