

Azure Language Services

By - Jayabharathi Hari
Deakin University
✉: s224643593@deakin.edu.au

Bot using Azure Cognitive Services Language Studio & Python SDK

Azure Image Analysis Introduction.....	1
Let's Build Bot using Azure Cognitive Services Language Studio & Python SDK.....	2
1.Create Language Resource.....	2
2. Launch Language Studio.....	5
3. Working in Language Studio using Python SDK and Azure Portal.....	8
3.1 Prerequisites in Jupyter Notebook:.....	8
3.2 Creating a project using python SDK in Language Studio.....	10
Code Explanation.....	12
3.3 Adding a knowledge base in azure portal.....	12
Manage Sources.....	12
Edit Knowledge Base Screen.....	13
3.4 Deploy knowledge base.....	15
4.Create a Bot.....	15
4.1 Creating resources for Bot.....	16
4.2 Test in Web Chat.....	20
4.3 Let's Connect with Channels(Telegram).....	21
I. Create a new Telegram bot with BotFather.....	22
II. Get the HTTP API/ Token.....	24
III. Access Token/Call API.....	26
IV. Test in Telegram.....	28
5.Deleting Resource Group.....	32
Reference.....	32

Azure Image Analysis Introduction ↗

Azure AI Language is a cloud-based service that provides Natural Language Processing (NLP) features for understanding and analyzing text. Use this service to help build intelligent applications using the web-based Language Studio, REST APIs, and client libraries.[1]

Below Listed are Azure AI Language Service Capabilities.[1]

- ✓ Azure AI Language capabilities
 - > Custom text classification
 - > Conversational language understanding
 - > Entity linking
 - > Language detection
 - > Key phrase extraction
 - > Named Entity Recognition (NER)
 - > Orchestration workflow
 - > Personally Identifiable Information (PII) detection
 - > Custom question answering
 - > Sentiment analysis and opinion mining
 - > Text Analytics for health
 - > Summarization

Let's Build Bot using Azure Cognitive Services Language Studio & Python SDK

1.Create Language Resource.

As shown below, we need to provide all required details such as Storage account , Language. Search service in order to create a language resource group.

Create Language

Basics Network Identity Tags Review + create

Unlock insights from unstructured text using advanced natural language processing. Use sentiment analysis to find out what customers think of your brand. Find topic-relevant phrases using key phrase extraction and identify the language of the text with language detection. Detect and categorize entities in your text with named entity recognition.

[Learn more](#)

Project Details

Subscription * Resource group * [Create new](#)

Instance Details

Region Name * Pricing tier *

[View full pricing details](#)

Custom question answering

Custom question answering lets you answer user's questions over your data corpus. You can extract questions and answers from your data, customize them and create a knowledge base. The knowledge base is stored in an Azure AI Search index in your own subscription.

[Learn more](#)

Azure search region Azure search pricing tier *

[View full Azure search pricing details](#)

Custom text classification, Custom named entity recognition, Custom summarization, Custom sentiment analysis & Custom Text Analytics for health

You need to create your own storage when using these customization features to host and upload all your training and labeled data and have it saved in your own Azure subscription. You get to associate only one storage account with one language resource that cannot be changed moving forward.

[Learn more](#)

New/Existing storage account * New storage account Existing storage account

Storage account name * Storage account type *

[Learn more about storage account types](#)

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Azure AI service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will

[Previous](#) [Next](#) [Review + create](#) [Give feedback](#)

We can see that deployment is done and now we can “go to resource group”

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "TextAnalyticsCreate-20240429223537". The deployment status is "Your deployment is complete". The deployment details table lists six resources created successfully:

Resource	Type	Status	Operation details
attachStorageAndS	Deployment	OK	Operation details
roleAssignmentsFo	Deployment	OK	Operation details
gkbotserviceazure	Azure AI services	OK	Operation details
gkbotserviceazure	Azure AI services	OK	Operation details
gkbotserviceazure-	Search service	Created	Operation details
deployStorage	Deployment	OK	Operation details

Next steps: [Go to resource group](#)

We were able to see language, search service and storage account resources got created.

The screenshot shows the Microsoft Azure Resource Group Overview page for the resource group "chatbotservicerg". The essentials section shows the following details:

- Subscription: Azure for Students
- Deployment: 4 Succeeded
- Location: East US

The resources section lists three resources:

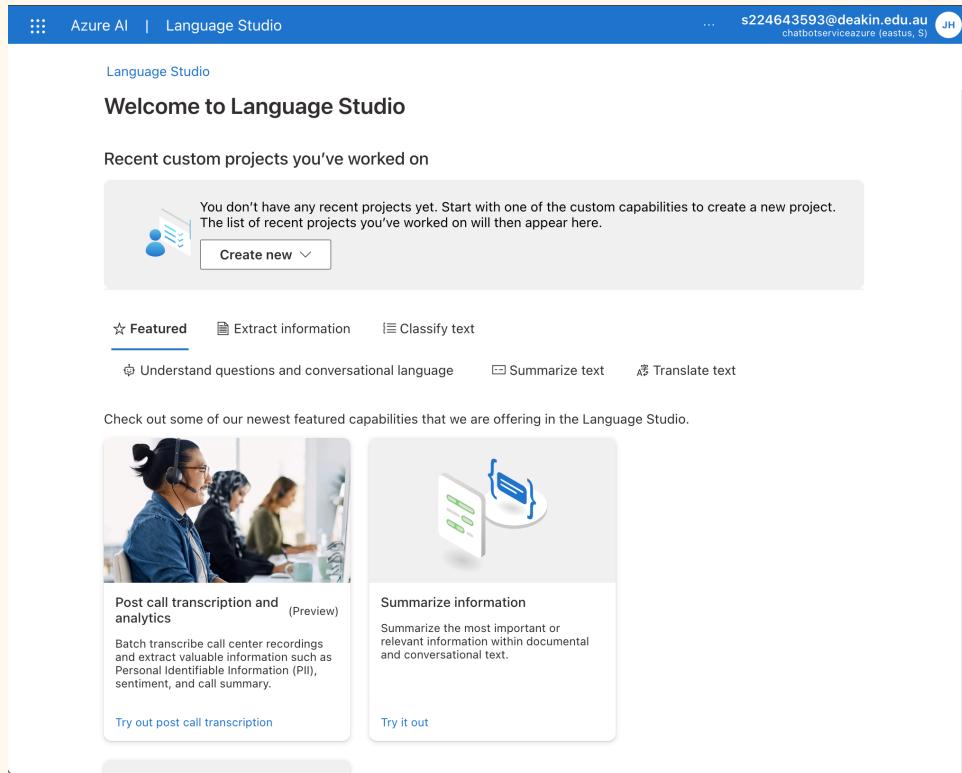
Name	Type	Location	...
gkbotserviceazure	Language	East US	...
gkbotserviceazure-asflubpta7bukpc	Search service	East US	...
gkbotstorage	Storage account	East US	...

2. Launch Language Studio.

Once we click on a language resource, we will come to the page below where we can launch the language studio.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure', a search bar, and user information ('s224643593@deakin.edu.au DEAKIN UNIVERSITY (DEAKIN)'). Below the navigation is a resource details card for 'chatbotserviceazure'. The card includes sections for Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. Under 'Overview', it shows the location as 'East US', subscription as 'Azure for Students', and endpoint as 'https://chatbotserviceazure.cognitiveservices.azure.com/'. The 'Get Started' tab is selected, leading to the 'Get Started with Language service' section. This section features three main steps: 'Discover', 'Develop', and 'Deploy', each with a corresponding icon and brief description. A note at the bottom states: 'Text Analytics has been rebranded and incorporated into Azure AI service for Language. [Learn More](#)'.

The screenshot shows the 'Language Studio' interface within the Azure AI service. The main page displays a welcome message and various features like 'Recent custom projects', 'Featured', and 'Translate text'. In the center, a modal dialog titled 'Select an Azure resource' is open. It asks for an 'Azure directory' (set to 'Deakin University') and an 'Azure subscription' (set to 'Azure for Students'). The 'Resource type' is set to 'Language'. A 'Resource name' input field contains 'gkbotserviceazure'. Below these fields are 'Pricing tier' (Standard S) and 'Managed identity' (Enabled). At the bottom of the modal are 'Done' and 'Cancel' buttons. The background of the page shows learning resources and a summary text feature.



3. Working in Language Studio using Python SDK and Azure Portal.

3.1 Prerequisites in Jupyter Notebook:

1. Install azure question answering library
2. Import the required Libraries.
3. Authenticate and connect to the Language Service client using endpoint and Key.

1. Install the necessary libraries:

```
In [1]: # Install the azure question answering library
!pip install azure-ai-language-questionanswering

Collecting azure-ai-language-questionanswering
  Downloading azure_ai_language_questionanswering-1.1.0-py3-none-any.whl.metadata (19 kB)
Collecting azure-core<2.0.0,>=1.24.0 (from azure-ai-language-questionanswering)
  Using cached azure_core-1.30.1-py3-none-any.whl.metadata (37 kB)
Collecting isodate<1.0.0,>=0.6.1 (from azure-ai-language-questionanswering)
  Using cached isodate-0.6.1-py2.py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: requests>=2.21.0 in /Applications/anaconda3/lib/python3.11/site-packages (from azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (2.31.0)
Requirement already satisfied: six>=1.11.0 in /Applications/anaconda3/lib/python3.11/site-packages (from azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (1.16.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /Applications/anaconda3/lib/python3.11/site-packages (from azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /Applications/anaconda3/lib/python3.11/site-packages (from requests>=2.21.0->azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /Applications/anaconda3/lib/python3.11/site-packages (from requests>=2.21.0->azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Applications/anaconda3/lib/python3.11/site-packages (from requests>=2.21.0->azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /Applications/anaconda3/lib/python3.11/site-packages (from requests>=2.21.0->azure-core<2.0.0,>=1.24.0->azure-ai-language-questionanswering) (2024.2.2)
  Downloading azure_ai_language_questionanswering-1.1.0-py3-none-any.whl (113 kB)
    113.1/113.1 kB 4.0 MB/s eta 0:00:00
Using cached azure_core-1.30.1-py3-none-any.whl (193 kB)
Using cached isodate-0.6.1-py2.py3-none-any.whl (41 kB)
Installing collected packages: isodate, azure-core, azure-ai-language-questionanswering
Successfully installed azure-ai-language-questionanswering-1.1.0 azure-core-1.30.1 isodate-0.6.1
```

2. Import the required libraries:

```
In [3]: # To read the secret keys for Authentication
import os
from dotenv import load_dotenv
from azure.core.credentials import AzureKeyCredential

# To create new project
from azure.ai.language.questionanswering.authoring import AuthoringClient

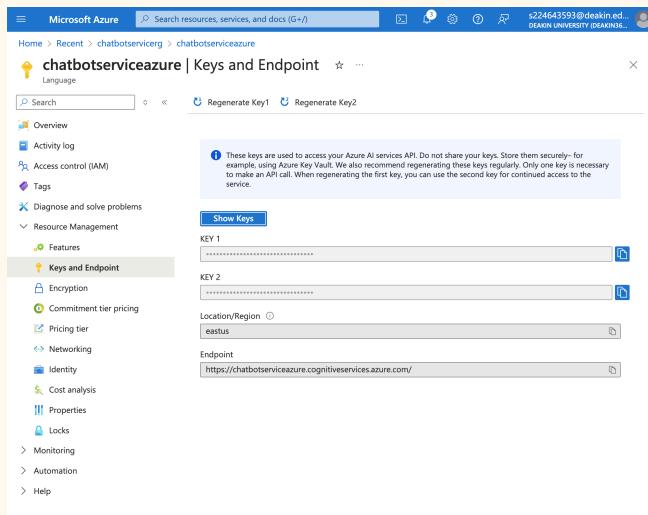
# to create a question-answering client, and to ask questions using the knowledge base
from azure.ai.language.questionanswering import QuestionAnsweringClient
from azure.ai.language.questionanswering import models as qna
```

3. Authenticate and connect to the Language Service client

```
In [5]: endpoint = "https://chatbotserviceazure.cognitiveservices.azure.com/"
key = "0ac2c13910dc4768888bca73762c5842"

# Please note, we will create two clients, one for creating the project (Authoring) and one for querying it (QuestionAnswering)

authoring_client = AuthoringClient(endpoint, AzureKeyCredential(key))
qna_client = QuestionAnsweringClient(endpoint, AzureKeyCredential(key))
```



3.2 Creating a project using python SDK in Language Studio.

I'm creating general knowledge bot so name the project as "GKBot".

```
authoring_client = AuthoringClient(endpoint, AzureKeyCredential(key))

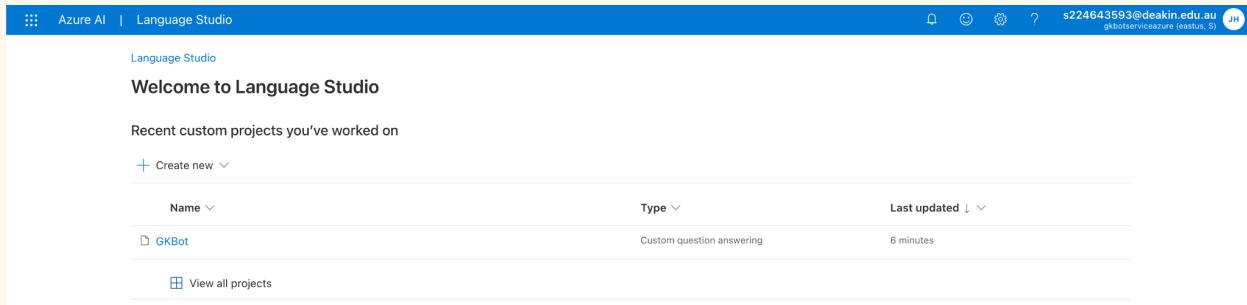
with authoring_client:

    # Step 1: create project

    print("\n***** Creating a new project *****")
    project_name = "GKBot"
    project = authoring_client.create_project(
        project_name=project_name,
        options={
            "description": "FAQs related to Azure (MS Learn and CLU)",
            "language": "en",
            "multilingualResource": True,
            "settings": {
                "defaultAnswer": "no answer"
            }
        })
    # Output 1: View the project details

    print("view created project info:")
    print("\tname: {}".format(project["projectName"]))
    print("\tlanguage: {}".format(project["language"]))
    print("\tdescription: {}".format(project["description"]))

***** Creating a new project *****
view created project info:
    name: GKBot
    language: en
    description: FAQs related to Azure (MS Learn and CLU)
```



Code Explanation

- Project Creation:** The code begins by creating a new project named "GKBot" using the `create_project` method of the "AuthoringClient" object. It provides some options for the

project, including a description, language (English), and settings such as the default answer if a question does not have a specific response.

2. **Output Project Details:** After creating the project, it prints out some details about the newly created project, including its name, language, and description.

3.3 Adding a knowledge base in azure portal

We can add a knowledge base as a source via url, file and chitchat in “Manage Sources”, I have added via file.

Manage Sources

The screenshot shows the Azure AI Language Studio interface. The top navigation bar includes 'Azure AI' and 'Language Studio'. The main title is 'Language Studio > Custom question answering > GKBot - Manage sources'. Below the title, there's a 'Manage sources' section with buttons for 'Add source', 'Edit name', 'Refresh URL', and 'Delete'. A message indicates '0 items in list' and provides a 'Filter' option. A central area features a decorative graphic of blue spheres. An 'Add files' dialog box is open in the foreground. The dialog has a header 'Add files' and a note: 'You can add up to 10 files at a time. Classification of file structure is auto-detected by default, but you can set your own file structure classification if preferred.' It contains fields for 'Source name' (set to 'Excel'), 'File name' (set to 'GKbot.xlsx'), and 'Classify file structure' (set to 'Auto-detect'). At the bottom of the dialog are 'Add all' and 'Cancel' buttons.

File is successfully uploaded in Manage Sources, below is the screenshot for the same.

A screenshot of the Azure AI Language Studio interface. The top navigation bar includes 'Azure AI | Language Studio', the user's email 's224643593@deakin.edu.au', and a profile icon. The left sidebar has a tree view with 'Language Studio', 'Custom question answering', 'Azure Search', 'GKBot' (selected), 'Manage sources' (selected), 'Edit knowledge base', 'Deploy knowledge base', 'Review suggestions', and 'Project settings'. The main content area is titled 'Manage sources' and shows a table with one item: 'GKbot.xlsx' (Source: Excel, Status: No). There are buttons for 'Add source', 'Edit name', 'Refresh URL', and 'Delete'. A search bar and filter button are also present.

Edit Knowledge Base Screen.

The Edit Knowledge Base Screen in Azure allows users to modify and manage the content and settings of their knowledge base projects.

Edit knowledge base

0 unstructured sources and 1 structured sources.
View sources

Hi Enable rich text Show context tree

Source: GKbot.xlsx

Answer

Edit answer

Hello, How may I help you

Alternate questions (1)

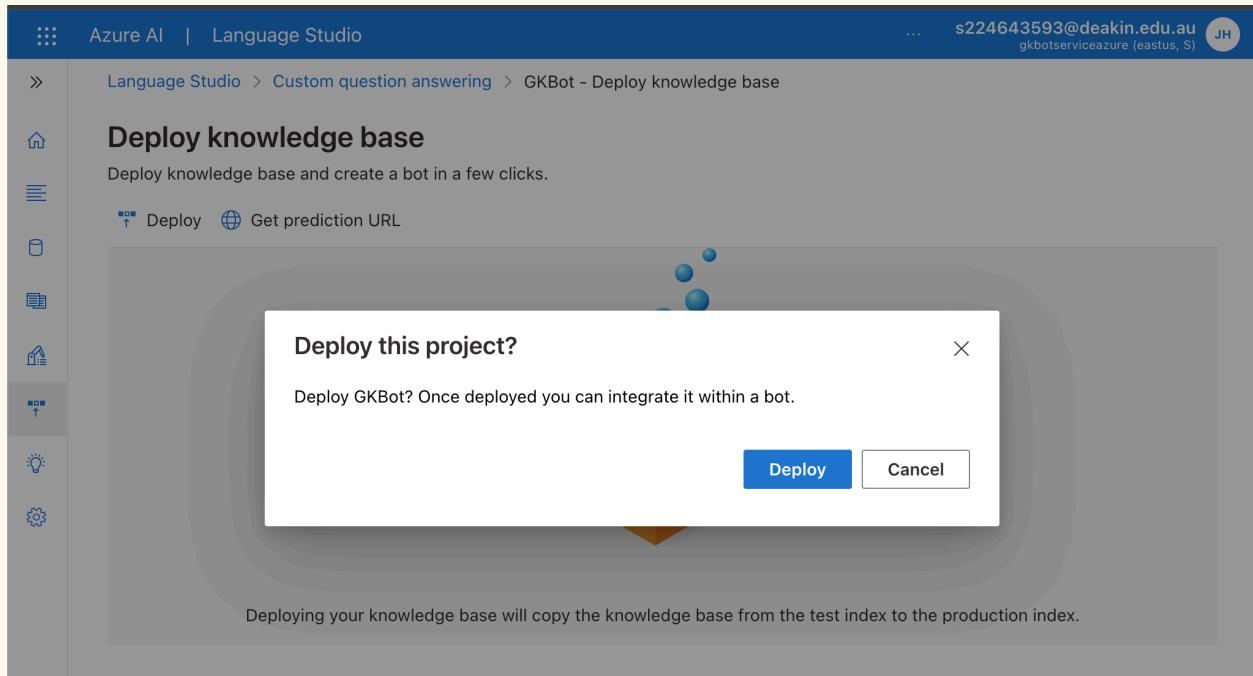
Follow up prompts (0)

Metadata (0)

Question	Answer
Hi	Hello, How may I help you
How are you	I'm doing well, Thanks for asking! How can I assist you today?
what is your name	As for my name, I'm GK bot, a language model developed using Microsoft Azu...
Who wrote "To Kill a Mockingbird"?	Harper Lee wrote "To Kill a Mockingbird."
What is the chemical symbol for gold?	The chemical symbol for gold is Au.
What is the capital of Japan?	The capital of Japan is Tokyo.
Who painted the Mona Lisa?	The Mona Lisa was painted by Leonardo da Vinci.
What is the tallest mammal on Earth?	The tallest mammal on Earth is the giraffe.
Which planet is known as the "Red Planet"?	Mars is known as the "Red Planet."
Who discovered penicillin?	Alexander Fleming is credited with the...

3.4 Deploy knowledge base

Deploying a knowledge base in Azure Language Studio involves making it accessible for querying through endpoints for integration into applications or services. I deployed my “GKbot” Project.



Deployment is completed.

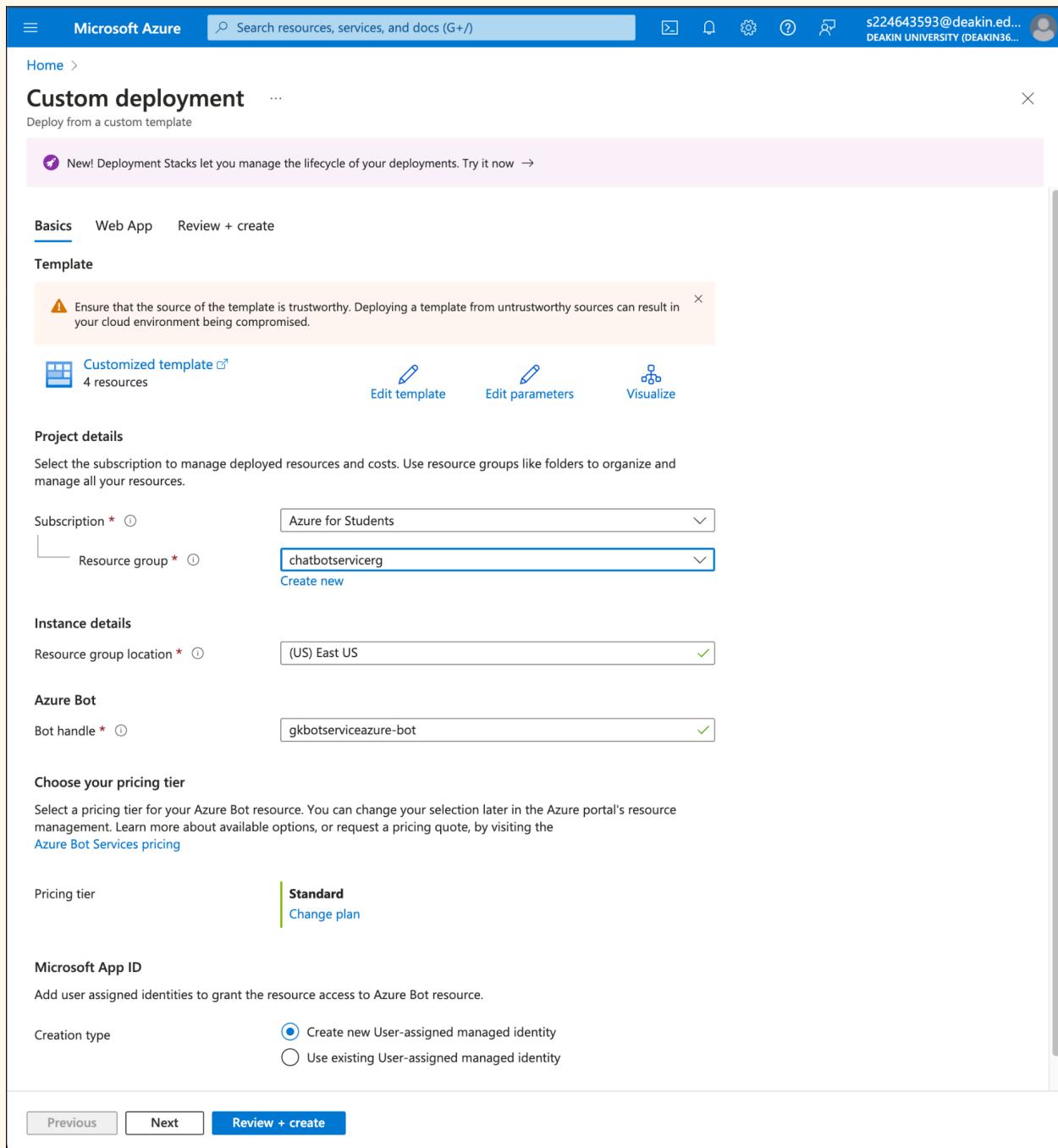
4.Create a Bot

Once Deployment is completed, we can see below screen where we would able to “Create a Bot”

Knowledge base status	
State:	① Deployed
Deployment Date:	② 4/30/2024
Deployment Time:	③ 8:52:49 AM
Resource:	gkbotserviceazure
Location:	eastus
Tier:	Standard (S)

4.1 Creating resources for Bot

While Creating resources , we need to make a connection with bot service and language service using the language resource key.



The screenshot shows the Microsoft Azure portal's "Custom deployment" wizard. The title bar says "Custom deployment" and "Deploy from a custom template". The "Basics" tab is selected. The "Template" section shows a warning about deploying from untrustworthy sources. It lists a "Customized template" with 4 resources, and buttons for "Edit template", "Edit parameters", and "Visualize". The "Project details" section shows the subscription set to "Azure for Students" and the resource group set to "chatbotservicerg". The "Instance details" section shows the resource group location set to "(US) East US" and the bot handle set to "gkbotserviceazure-bot". The "Choose your pricing tier" section has "Standard" selected. The "Microsoft App ID" section is present. At the bottom, there are "Previous", "Next", and "Review + create" buttons.

Custom deployment ...

Deploy from a custom template

⚠ Changes on this step may reset later selections you have made. Review all options prior to deployment.

Basics **Web App** **Review + create**

App Service

App name * .azurewebsites.net

SDK language selection *

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.
[Learn more](#)

Creation type Create new app service plan Use existing app service plan

Info The pricing tier currently defaults to 'S1 Standard'. It can be modified by visiting the app service plan resource page once the resource has been created, or you can choose an existing plan in your subscription.

App Settings

In App Service, these app settings are variables passed as environment variables to the bot code.

Language Resource Key *
Language project name	GKBot
Language service endpoint hostname	https://gkbotserviceazure.cognitiveservices.azure.com
Language service details	
Subscription Id	/subscriptions/8d5ceb22-44c7-4367-bfc6-07378096e055
Resource Group Name	chatbotservicerg
Account Name	gkbotserviceazure

Previous **Next** **Review + create**

We can view all resources including “Azure bot” listed in the main resource group.

Microsoft Azure Search resources, services, and docs (G+) s224643593@deakin.edu... DEAKIN UNIVERSITY (DEAKIN36...)

Home > chatbotservicerg Resource group

[Search](#) [Create](#) [Manage view](#) [Delete resource group](#) [Refresh](#) [Export to CSV](#) [Open query](#) [Assign tags](#) [Move](#)

[Overview](#) [Essentials](#) [JSON View](#)

[Activity log](#) [Subscription \(move\)](#) [Deployments](#)
[Access control \(IAM\)](#) [Azure for Students](#) [5 Succeeded](#)
[Tags](#) [Subscription ID](#) [Location](#)
[Resource visualizer](#) [06fc316e-08b3-4768-9e29-bb9cc1fbaf53](#) [East US](#)
[Events](#) [Tags \(edit\)](#)
[Settings](#) [Add tags](#)
[Cost Management](#)
[Monitoring](#)
[Automation](#)
[Help](#)

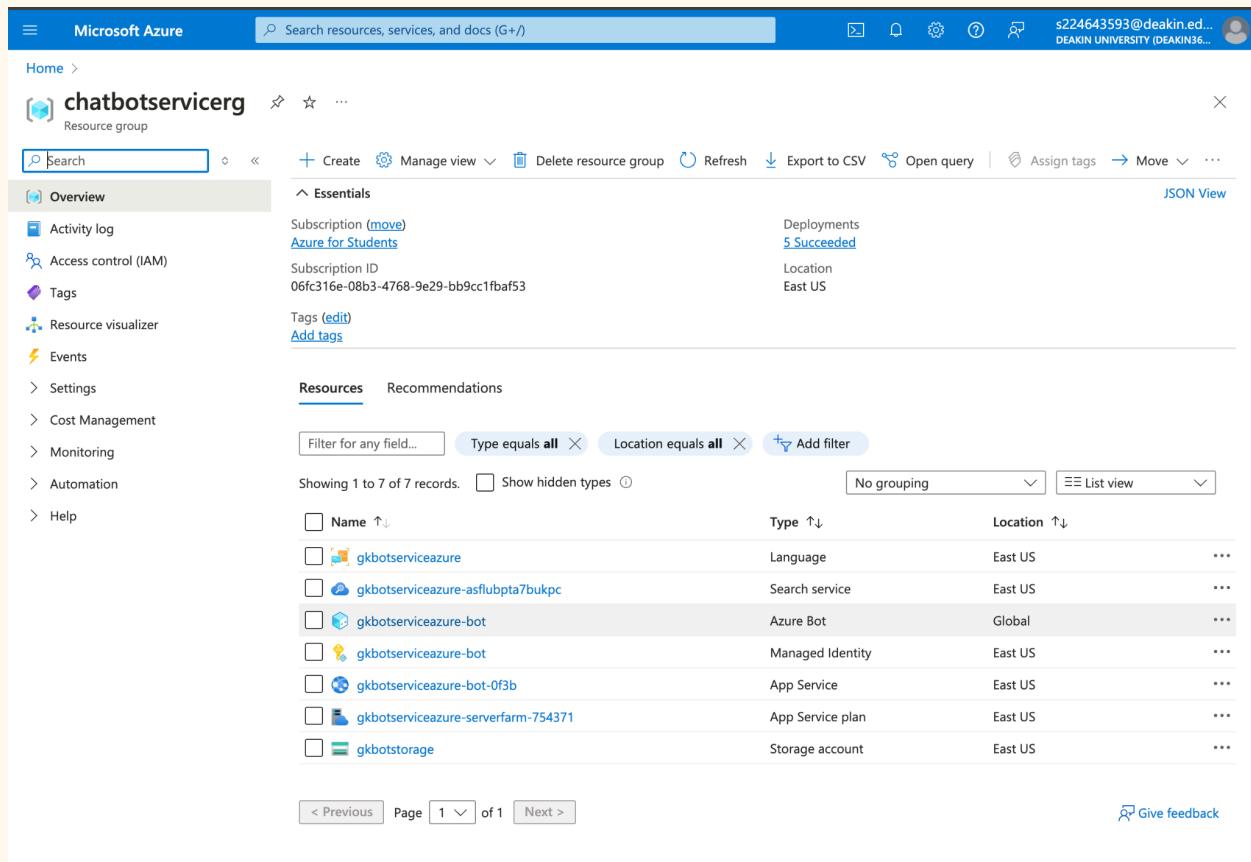
[Resources](#) [Recommendations](#)

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 7 of 7 records. Show hidden types No grouping List view

Name	Type	Language	Location	...
gkbotserviceazure	Search service	East US	East US	...
gkbotserviceazure-asflubpta7bukpc	Azure Bot	Global	Global	...
gkbotserviceazure-bot	Managed Identity	East US	East US	...
gkbotserviceazure-bot-0f3b	App Service	East US	East US	...
gkbotserviceazure-serverfarm-754371	App Service plan	East US	East US	...
gkbotstorage	Storage account	East US	East US	...

< Previous Page 1 of 1 Next > [Give feedback](#)



4.2 Test in Web Chat

Microsoft Azure

Search resources, services, and docs (G+)

Home > chatbotservicerg >

gkbotserviceazure-bot Azure Bot

Overview

Activity log

Access control (IAM)

Tags

Settings

Monitoring

Automation

Help

Download bot source code Refresh Move Delete

WebAppLinked : true

Build enterprise-grade conversational AI

Develop enterprise-grade, conversational AI experiences, while maintaining control of your data. Edit your bot in SDK, host your bot in any environment and enable customers from a variety of channels, such as your app or website, Direct Line Speech, Microsoft Teams and more. [Learn more](#)

Get started with the Bot Framework

Build, test, deploy, and manage intelligent bots, all in one place. The Bot Framework includes a modular and extensible SDK for building bots, as well as tools, templates, and related AI services.

Note: For User-Assigned Managed Identity and Single Tenant app, BotFramework SDK (C# or Javascript) version 4.15.0 or higher is needed for these app types. [Learn the Bot Framework](#)

Test and refine your bot

Refined and debug locally with [Emulator](#), and test your bot online in [Web Chat](#). Learn more about [testing and debugging](#).

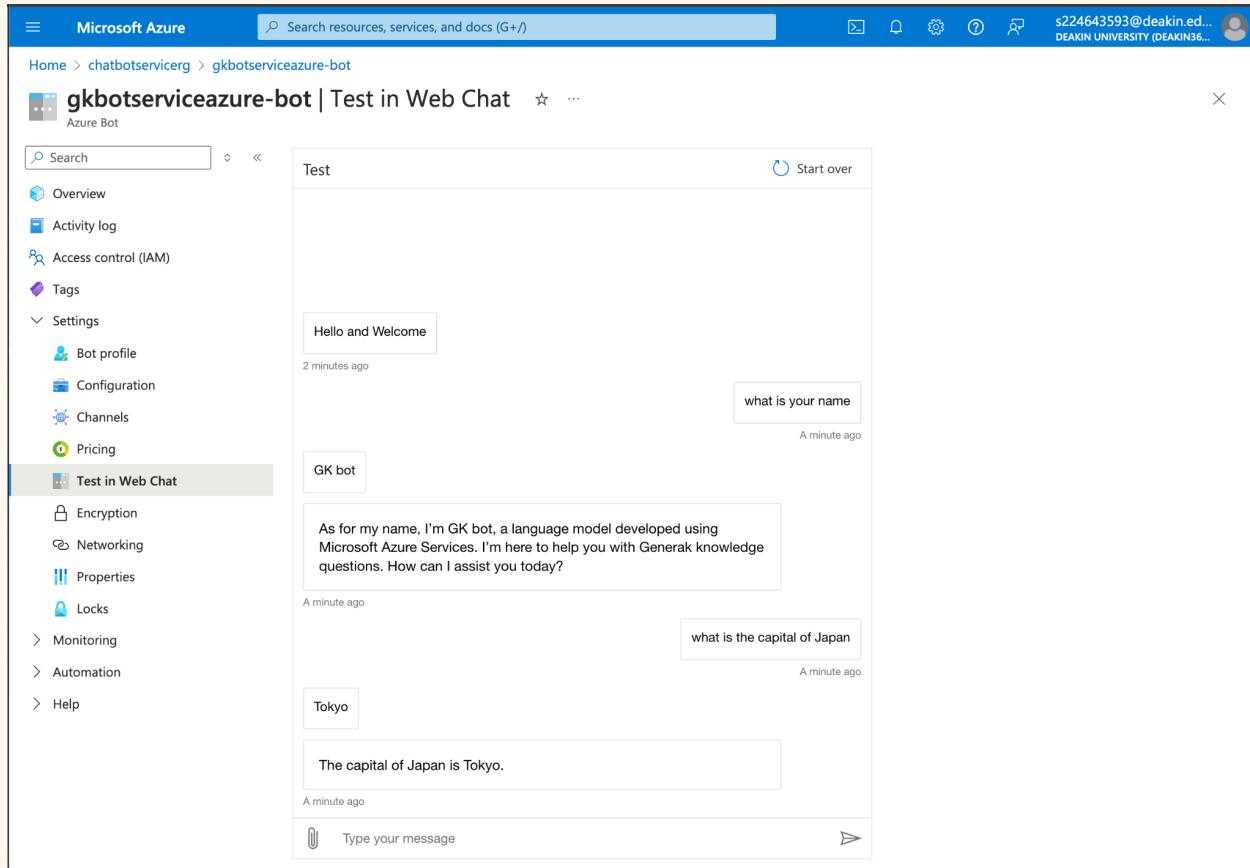
Test in Web Chat

Download bot source code

Publish to Azure

When you are ready, publish your bot to Azure and [connect to channels](#). Learn about publishing directly to Azure and [continuous deployment](#).

Connect to channels



4.3 Let's Connect with Channels(Telegram)

Channels in Azure Language Services refer to communication interfaces like web chat or messaging platforms like facebook Messenger, Telegram, Microsoft Teams, Skype etc. It facilitates interaction with deployed language models.

The screenshot shows the Microsoft Azure portal interface for managing a bot named "gkbotserviceazure-bot". The left sidebar has a tree view with "Channels" selected. The main content area displays the bot's current connections and a list of available channels. The "Available Channels" section includes Alexa, Communication Services - Chat, Direct Line Speech, Email, Facebook, GroupMe, and LINE.

Channel	Health status	Details	Actions
Direct Line	Healthy	REST API for communicating directly with a bot	
Web Chat	Healthy	Embeddable Web Chat control	

Channel	Details
Alexa	Alexa Channel
Communication Services - Chat	Communication Services - Chat Channel
Direct Line Speech	Direct Line Speech Channel
Email	O365 Email Channel
Facebook	Support for Text Messaging via Facebook
GroupMe	GroupMe Channel
LINE	Support for LINE Channel

Let's Connect with Telegram Channel.

I. Create a new Telegram bot with BotFather

We can follow the [Microsoft AI Bot Service](#) instructions to connect our language model “GKbot” with Telegram.

before you begin.

- An existing bot published to Azure.
- A device with [Telegram](#) installed and a Telegram account.

Create a new Telegram bot with BotFather

Create a Telegram bot with BotFather before connecting your bot to Telegram.

- Start a new conversation with the [BotFather](#).

Additional resources

Training
[Module: Channelized Azure Health Bot - Training](#)
This module demonstrates how to make a bot available on the Microsoft Teams channel. It also shows how to build a...

Documentation
[Implement channel-specific functionality in Bot Framework SDK - Bot Service](#)
Learn how to implement channel-specific functionality using the Bot Framework SDK for .NET. You can do so by passing...

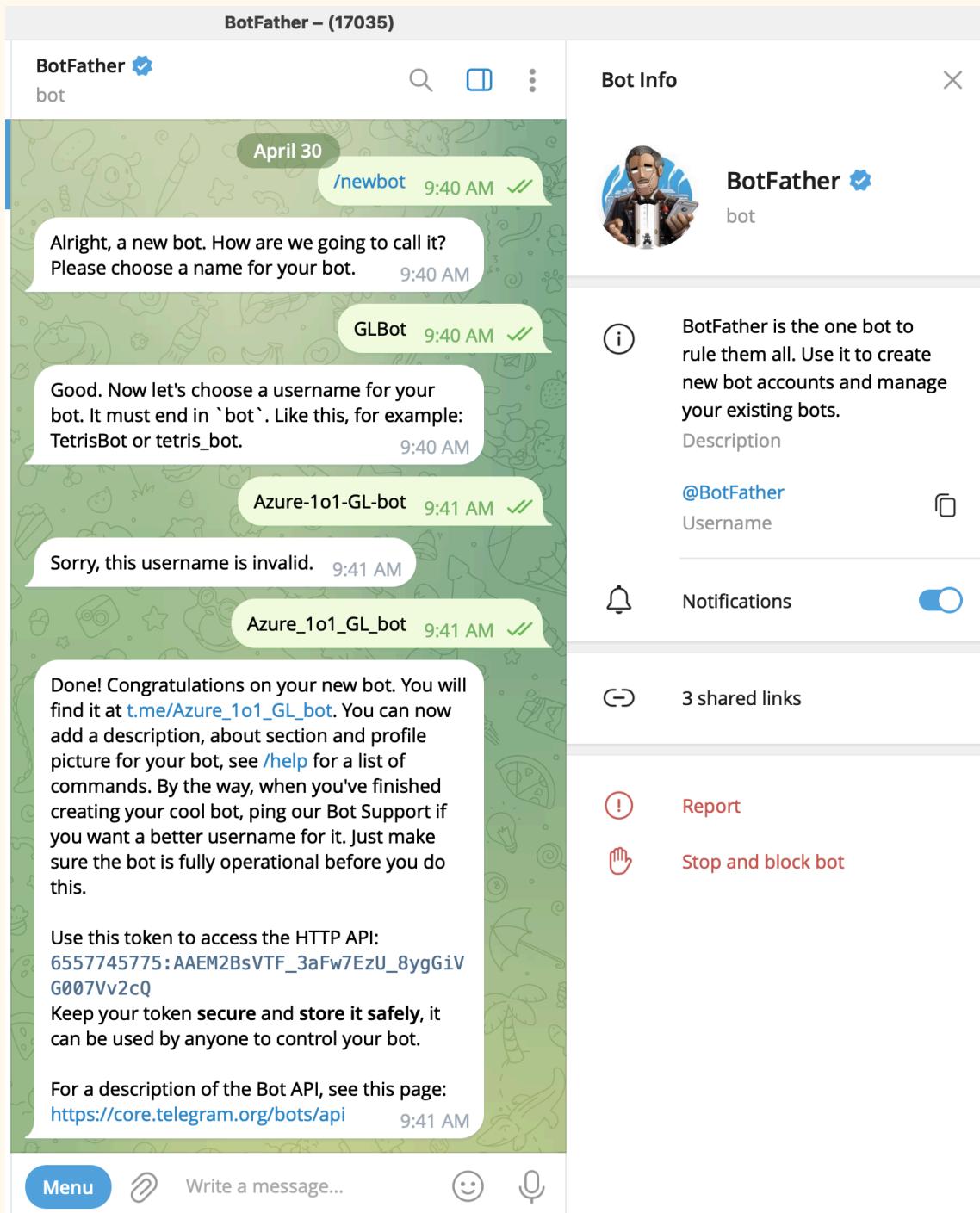
[Connect a bot to WeChat - Bot Service](#)
Learn how to configure a bot's connection to WeChat.

[Connect a Bot Framework bot to Facebook - Bot Service](#)
Learn how to configure bots to connect to Facebook Messenger and Facebook Workplace and communicate with users...

[Show 5 more](#)

II. Get the HTTP API/ Token

Link navigates to the “BotFather” in telegram desktop App. Once its validates the bot name, “BotFather” will provide the HTTP API/Token which we needs to call/link in Azure channel Connection. Below is the screenshot where we can see the HTTP API/Token.



III. Access Token/Call API

Provide the token in telegram credentials - Azure Portal.

The screenshot shows the Microsoft Azure portal interface for managing a bot named 'gkbotserviceazure-bot'. The left sidebar navigation menu includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings' (which is expanded to show 'Bot profile', 'Configuration', 'Channels', 'Pricing', 'Test in Web Chat', 'Encryption', 'Networking', 'Properties', 'Locks'), 'Monitoring', and 'Automation'. The 'Channels' option under 'Settings' is currently selected. The main content area is titled 'gkbotserviceazure-bot | Telegram' and displays instructions for connecting a bot to a Telegram channel. It features a 'Search' bar, a toolbar with 'Close', 'Delete channel', 'Refresh', and 'Feedback' buttons, and a note about configuring communication between the bot and end users. A section titled 'Telegram credentials' contains a field labeled 'Access Token *' with a red asterisk, which is filled with a series of asterisks ('*****'). At the bottom are 'Apply' and 'Discard changes' buttons, and a toggle switch labeled 'Enabled' which is set to 'Enabled'.

Once we initialized the telegram token , we can see that Telegram channel is enabled and also available in the channels page.

The screenshot shows the Microsoft Azure portal interface for managing a bot. The left sidebar lists various settings like Overview, Activity log, Access control (IAM), Tags, Settings (with Bot profile and Configuration), Channels (selected), Pricing, Test in Web Chat, Encryption, Networking, Properties, Locks, Monitoring, and Automation. The main content area is titled 'gkbotserviceazure-bot | Telegram' and shows instructions for connecting the bot to a Telegram channel. It includes a 'Telegram credentials' section with an 'Access Token *' input field containing a redacted token. Below this is a 'Notifications' panel with a message stating 'Telegram channel enabled' and 'The channel's set-up is complete.' A timestamp 'a few seconds ago' is shown. At the bottom are 'Apply' and 'Discard changes' buttons, and a toggle switch labeled 'Enabled'.

All three Channels are showing as “healthy” in connection.

The screenshot shows the Microsoft Azure portal interface for managing a bot. The left sidebar lists various settings like Overview, Activity log, Access control (IAM), Tags, Settings (with Bot profile and Configuration), Channels (selected), Pricing, Test in Web Chat, Encryption, Networking, Properties, Locks, Monitoring, and Automation. The main content area is titled 'gkbotserviceazure-bot | Channels' and shows a message: 'You are using the updated channels page. Let us know what you think by providing feedback'. Below this, it states 'This bot is connected with the following channels.' A table lists the connected channels:

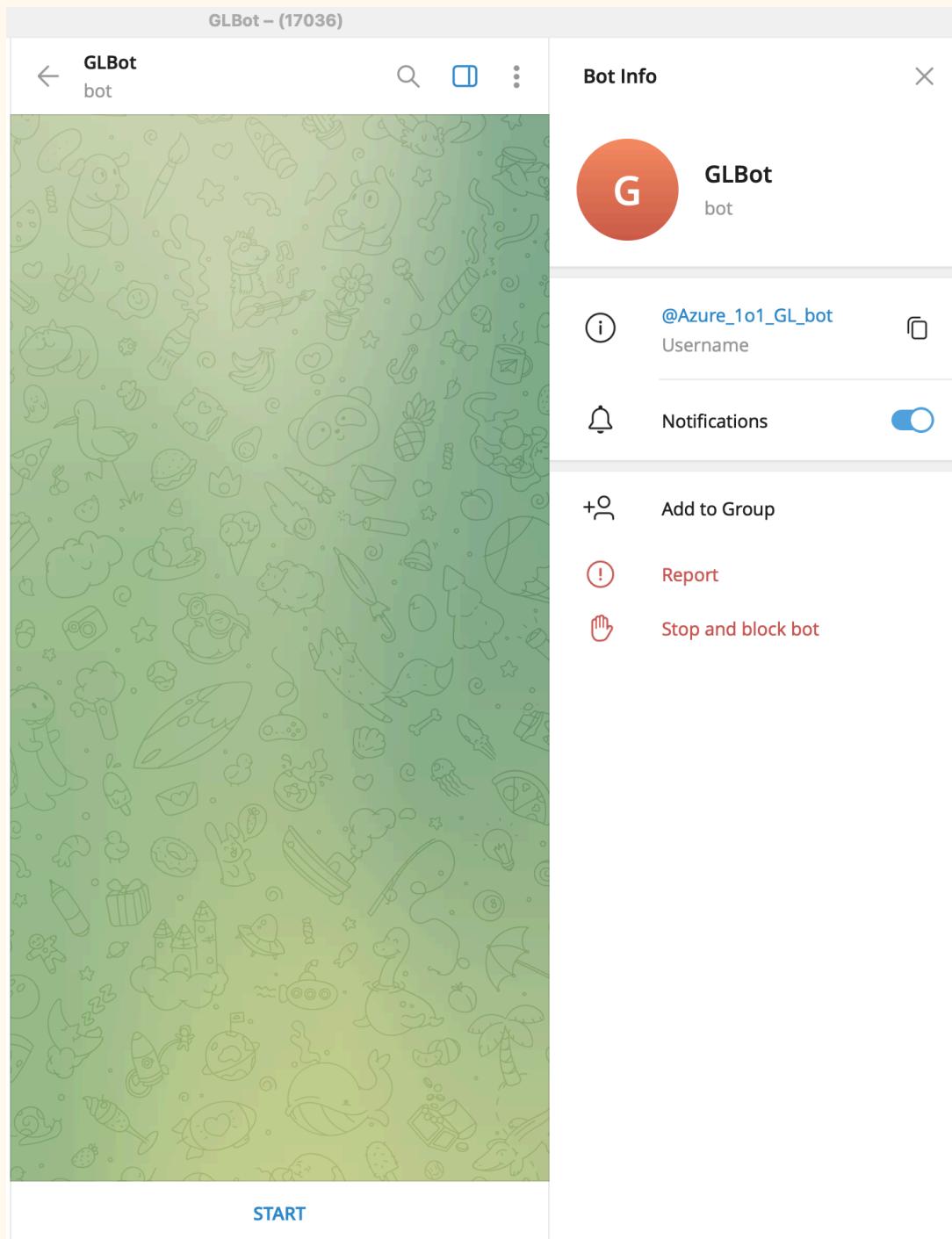
Channel	Health status	Details	Actions
Direct Line	Healthy	REST API for communicating directly with a bot	
Telegram	Healthy	Telegram Channel	Open in Telegram
Web Chat	Healthy	Embeddable Web Chat control	

Below the table, there is an 'Available Channels' section with a table:

Channel	Details
Alexa	Alexa Channel

IV. Test in Telegram

Now. Bot is live in Telegram.



And our GK bot is working perfectly in Telegram.

GLBot – (17036)

GLBot bot

Hi 9:54 AM ✓

Hello, How may I help you 9:54 AM

What is your Name 9:54 AM ✓

GK bot 9:54 AM

What is capital of Japan 9:54 AM ✓

Tokyo 9:54 AM

Who painted the Mona Lisa? 9:55 AM ✓

Leonardo da Vinci 9:55 AM

What is the tallest mammal on Earth? 9:55 AM ✓

giraffe 9:55 AM

Who discovered penicillin? 9:55 AM ✓

Alexander Fleming 9:55 AM

What is the chemical formula for water? 9:55 AM ✓

H₂O 9:55 AM

What is the chemical symbol for gold? 9:56 AM ✓

Au 9:56 AM

Who was the first person to step on the moon? 9:56 AM ✓

Neil Armstrong 9:56 AM

Write a message...

Bot Info

G GLBot bot

@Azure_1o1_GL_bot Username

Notifications

Add to Group

Report

Stop and block bot

5. Deleting Resource Group.

GK bot Model is working fine in telegram bot and tested in web chat, now let's delete Resource Group for Cost Management.

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

chatbotservicerg

Dependent resources to be deleted (7)

All dependent resources, including hidden types, are shown

Name	Resource type
gkbotserviceazure	Azure AI services
gkbotserviceazure-asflubpta7bukpc	Search service
gkbotserviceazure-bot	Managed Identity
gkbotserviceazure-bot	Azure Bot
gkbotserviceazure-bot-0f3b	App Service
gkbotserviceazure-serverfarm-754371	App Service plan
gkbotstorage	Storage account

Enter resource group name to confirm deletion *

chatbotservicerg

Delete Cancel

Reference

[1] : Microsoft Azure Portal. Link provided.