NAME: ANGEL TWUMASI

INDEX: 9414119

REF: 20648866

A Chatbot with Microsoft Azure

1. Introduction:

A chatbot is an application that can initiate and continue a conversation using auditory and/or textual methods as a human would do. A chatbot can be either a simple rule-based engine or an intelligent application leveraging Natural Language Understanding. Many organizations today have started using chatbots extensively. Chatbots are becoming famous as they are available 24*7, provide a consistent customer experience, can handle several customers at a time, are cost-effective and hence, result in a better overall customer experience.

1.1 Uses

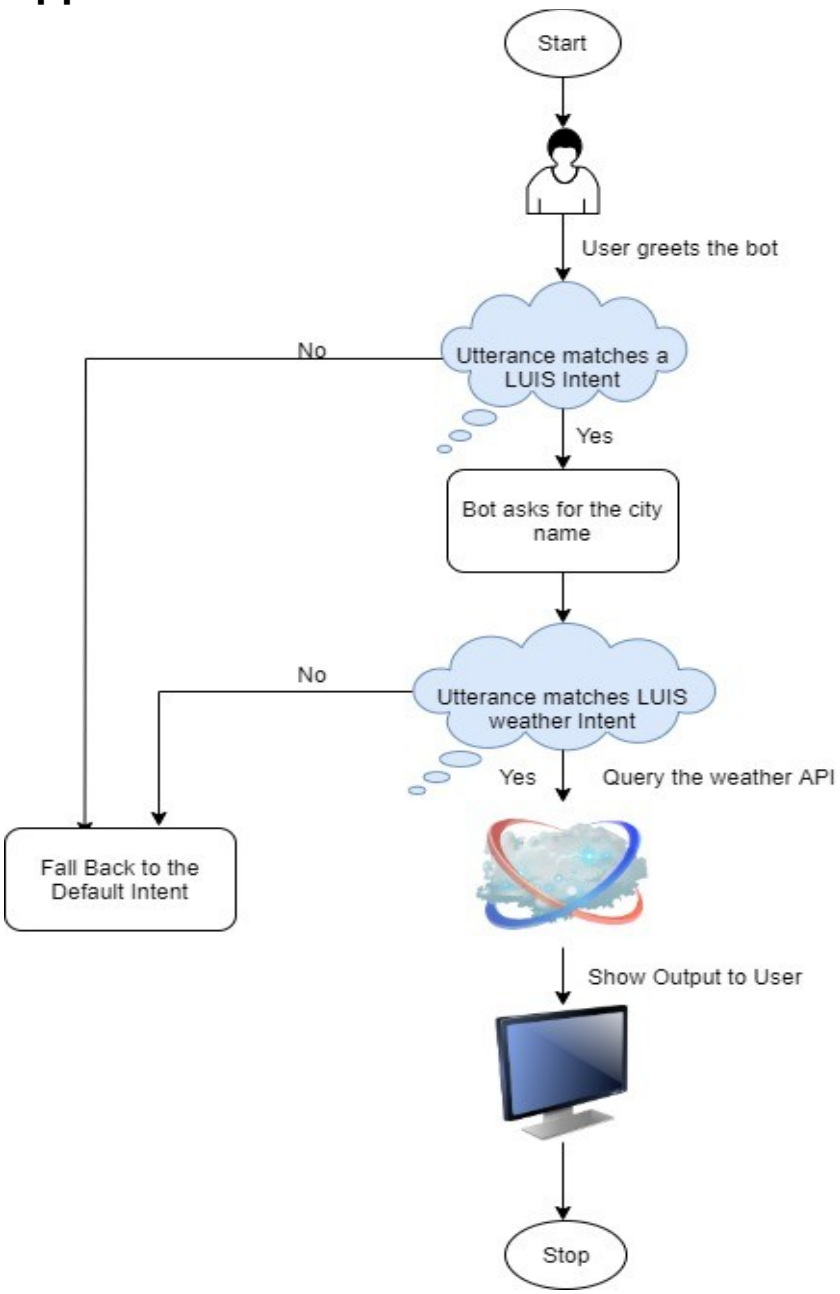
- Customer support
- Frequently Asked Questions
- Addressing Grievances
- Appointment Booking
- Automation of routine tasks
- Address a query

2. Prerequisites

The prerequisites for developing and understanding a chatbot using Microsoft Azure are:

- An Azure account.
- A fundamental understanding of python and flask

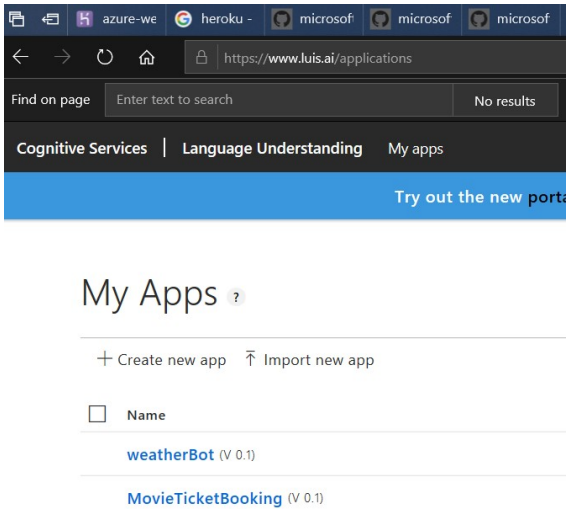
3. Application Architecture



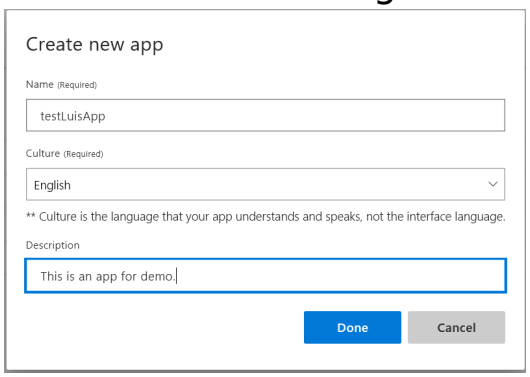
4. Implementation

4.1 Creating a LUIS App

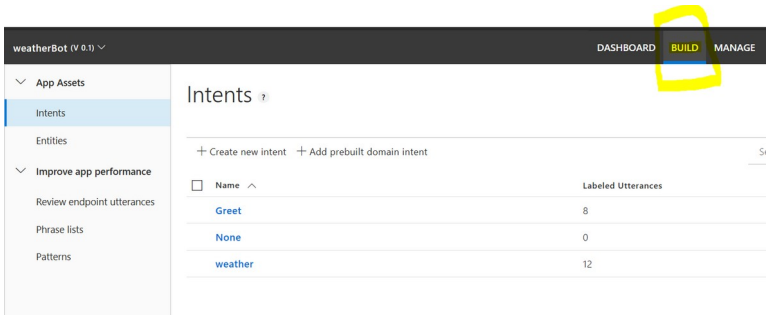
- Go to https://www.luis.ai and create an account if you already don't have one.
- Click on 'create new app' to create a new app by as shown:



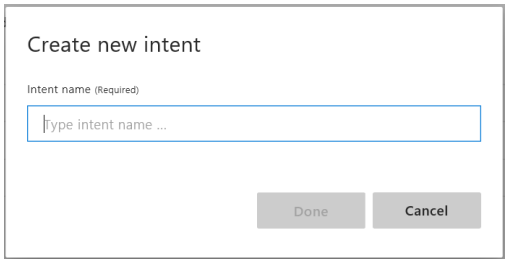
- Provide the following details and click ‘Done.



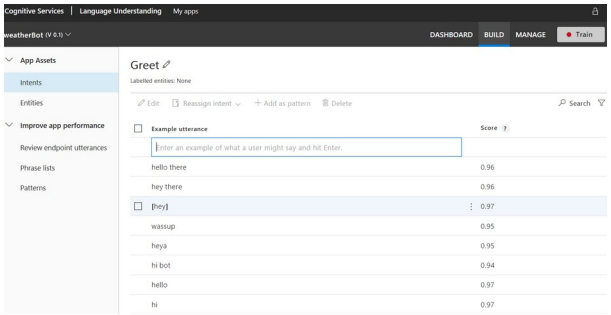
- Once created, open your app, select build, and click ‘Create new intent’ to create a new intent.



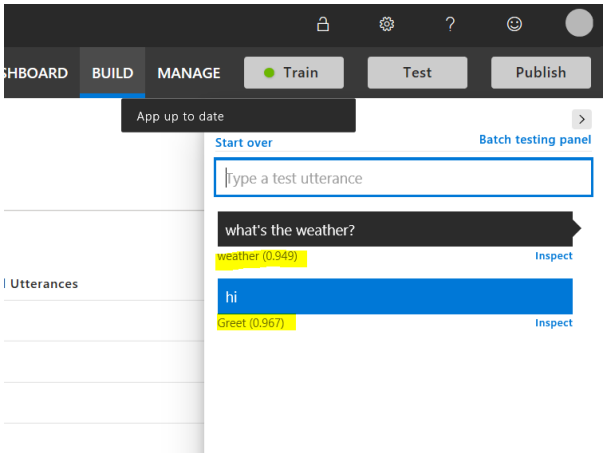
- Enter the name of the intent.



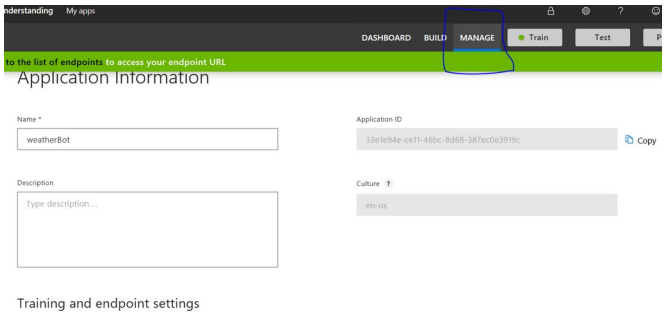
- Enter the user utterances and then click ‘train’ to train the LUIS app.



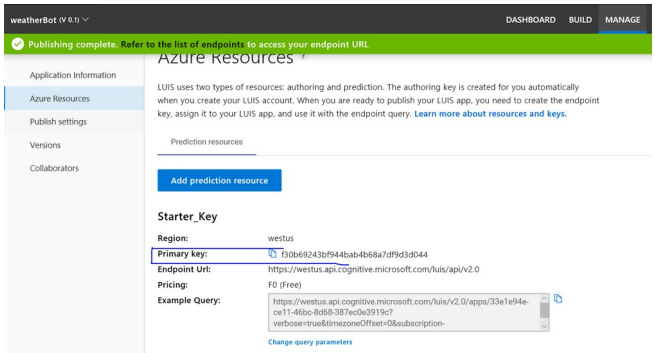
- Click ‘Test’ to test the intent and see the confidence of the app for various utterances.



- If satisfied with the test, click ‘publish’ and select the environment to make the LUIS app ready for consumption.
- Go to the *Manage* section of the published app and copy the *Application ID*. It will act as the ‘LUIS App ID.’



- Go to the *Azure Resources* section and copy the *Primary key*. It will serve as the LUIS API KEY.



4.2 Create a python app

4.2.1 Subscribing to the weather API

- Go to https://home.openweathermap.org/, sign in/signup, and create an API Key for calling the *current weather data* API.
- This will act as the *weather_api_key*.

4.2.2 App creation and Integration with LUIS

- Create a folder for your chatbot called azurePythonBot.
- Open the folder through pycharm.
- Create a file called app.py and put the following code.

```
from flask import Flask, request, Response
from botbuilder.core import BotFrameworkAdapter,
BotFrameworkAdapterSettings,
ConversationState,MemoryStorage
from botbuilder.schema import Activity
import asyncio
from luis.luisApp import LuisConnect
import os
from logger.logger import Log


app = Flask(__name__)
loop = asyncio.get_event_loop()

bot_settings = BotFrameworkAdapterSettings("", "")
bot_adapter = BotFrameworkAdapter(bot_settings)

#CON_MEMORY = ConversationState(MemoryStorage())
luis_bot_dialog = LuisConnect()


@app.route("/api/messages", methods=["POST"])
def messages():
    if "application/json" in request.headers["content-
type"]:
        log=Log()
        request_body = request.json
        user_says = Activity().deserialize(request_body)
```

```
log.write_log(sessionID='session1',log_message="user
says: "+str(user_says))
        authorization_header =
(request.headers["Authorization"] if "Authorization" in
request.headers else "")

        async def call_user_fun(turncontext):
            await luis_bot_dialog.on_turn(turncontext)

        task = loop.create_task(
            bot_adapter.process_activity(user_says,
authorization_header, call_user_fun)
        )
        loop.run_until_complete(task)
        return ""
    else:
        return Response(status=406)  # status for Not
Acceptable




if __name__ == '__main__':
    #app.run(port= 3978)
    app.run()
```

- Create a folder logger and create logger.py inside as shown below:

```
from datetime import datetime
class Log:
    def __init__(self):
        pass

    def write_log(self, sessionID, log_message):
        self.file_object =
open("conversationLogs/"+sessionID+".txt", 'a+')
        self.now = datetime.now()
        self.date = self.now.date()
        self.current_time = self.now.strftime("%H:%M:%S")
        self.file_object.write(
            str(self.date) + "/" + str(self.current_time)
+ "\t\t" + log_message + "\n")
        self.file_object.close()
```

- Create a folder luis and create luisApp.py inside as shown below:

```
from botbuilder.core import TurnContext,ActivityHandler
from botbuilder.ai.luis import
LuisApplication,LuisPredictionOptions,LuisRecognizer
import json
from weather.weatherApp import WeatherInformation
from config.config_reader import ConfigReader
from logger.logger import Log
```

```
class LuisConnect(ActivityHandler):
    def __init__(self):
        self.config_reader = ConfigReader()
        self.configuration =
self.config_reader.read_config()

self.luis_app_id=self.configuration['LUIS_APP_ID']
        self.luis_endpoint_key =
self.configuration['LUIS_ENDPOINT_KEY']
        self.luis_endpoint =
self.configuration['LUIS_ENDPOINT']
        self.luis_app =
LuisApplication(self.luis_app_id,self.luis_endpoint_key,s
elf.luis_endpoint)
        self.luis_options =
LuisPredictionOptions(include_all_intents=True,include_in
stance_data=True)
        self.luis_recognizer =
LuisRecognizer(application=self.luis_app,prediction_optio
ns=self.luis_options,include_api_results=True)
        self.log=Log()


    async def
on_message_activity(self,turn_context:TurnContext):
        weather_info=WeatherInformation()
        luis_result = await
self.luis_recognizer.recognize(turn_context)
        result = luis_result.properties["luisResult"]
        json_str =
json.loads((str(result.entities[0])).replace("'", "\""))

weather=weather_info.get_weather_info(json_str.get('entit
y'))

self.log.write_log(sessionID='session1',log_message="Bot
Says: "+str(weather))
        await turn_context.send_activity(f"{weather}")
```

- Create a folder weather and create weatherApp.py inside it as shown below:

```
import pyowm
from config.config_reader import ConfigReader

class WeatherInformation():
    def __init__(self):
        self.config_reader = ConfigReader()
        self.configuration =
self.config_reader.read_config()
        self.owmapikey =
self.configuration['WEATHER_API_KEY']
        self.owm = pyowm.OWM(self.owmapikey)

    def get_weather_info(self,city):
        self.city=city
```

```
        observation = self.owm.weather_at_place(city)
        w = observation.get_weather()
        latlon_res = observation.get_location()
        lat = str(latlon_res.get_lat())
        lon = str(latlon_res.get_lon())

        wind_res = w.get_wind()
        wind_speed = str(wind_res.get('speed'))

        humidity = str(w.get_humidity())

        celsius_result = w.get_temperature('celsius')
        temp_min_celsius =
str(celsius_result.get('temp_min'))
        temp_max_celsius =
str(celsius_result.get('temp_max'))

        fahrenheit_result =
w.get_temperature('fahrenheit')
        temp_min_fahrenheit =
str(fahrenheit_result.get('temp_min'))
        temp_max_fahrenheit =
str(fahrenheit_result.get('temp_max'))
        self.bot_says = "Today the weather in " + city
+".\n Maximum Temperature :"+temp_max_celsius+ " Degree
Celsius"+".\n Minimum Temperature :"+temp_min_celsius+ "
Degree Celsius" +": \n" + "Humidity :" + humidity + "%"
        return self.bot_says
```

- Create a file config.ini and put the following details:

```
[DEFAULT]
WEATHER_API_KEY=119242c426975bc98ee4f259b9551823
LUIS_APP_ID=33e1e94e-ce11-46bc-8d68-387ec0e3919c
LUIS_ENDPOINT_KEY=f30b69243bf944bab4b68a7df9d3d044
LUIS_ENDPOINT=https://westus.api.cognitive.microsoft.com/
```

- Create a folder config and and create config_reader.py inside as shown below:

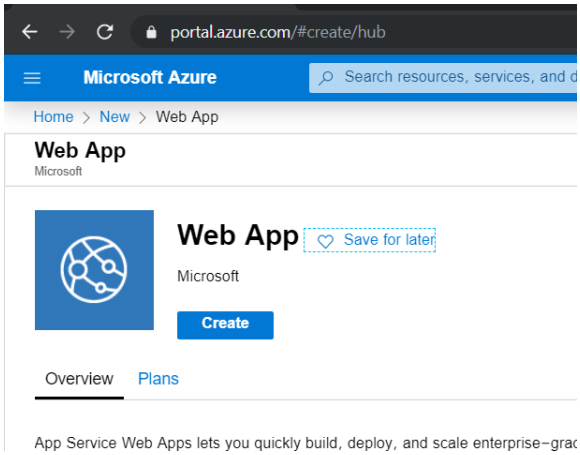
```
import configparser

class ConfigReader:
    def __init__(self):
        self.filename = 'config.ini'
    def read_config(self):
        self.config = configparser.ConfigParser()
        self.config.read(self.filename)
        self.configuration=self.config['DEFAULT']
        return self.configuration
```

NOTE: You could use Bot Emulator to test the application

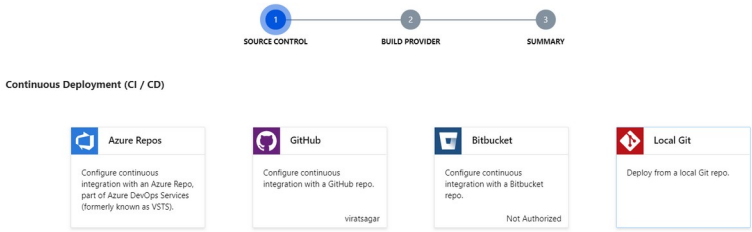
4.3 Deploy to azure

- Go to the Azure account and create a web app.

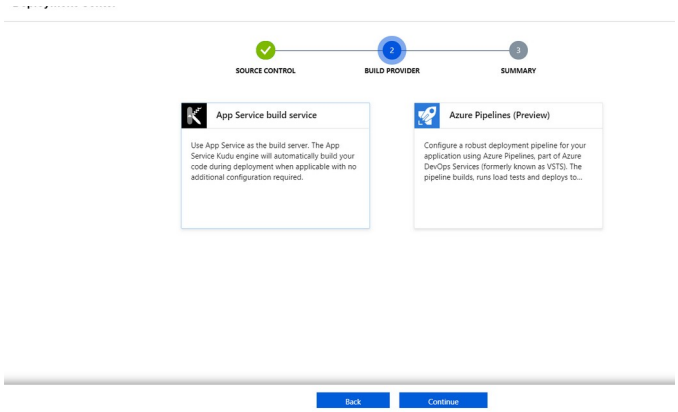


- Provide the app name, resource group(create new if necessary), runtime stack(Python <version>), region, select the 1 GB size, which is free to use. Click *Review+create* to create the web app.

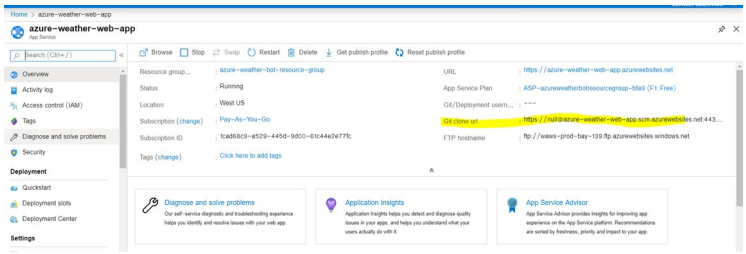
- Once the deployment is completed, open the app and go to the 'Deployment Center' option. Select 'local git' for source control and click continue.



- Select the kudo 'App service build provider' as the build provider and click continue.

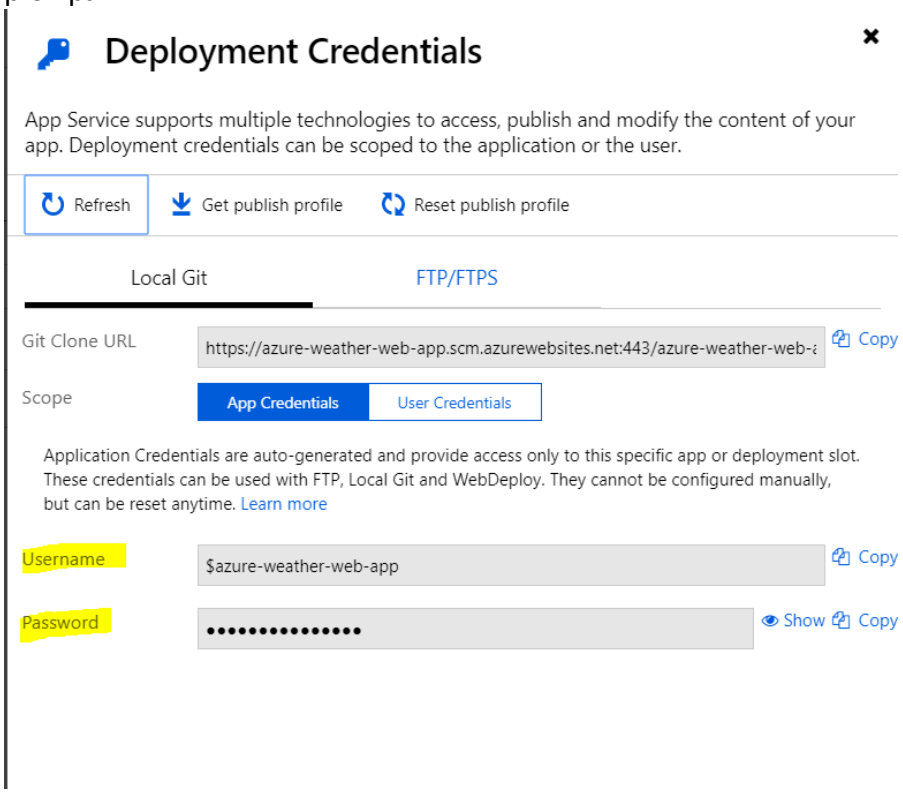


- Click 'Finish' to complete the setup.
- Go to the overview section of the app, and the Git link now will be visible.



- Open a command prompt and navigate to your project folder.

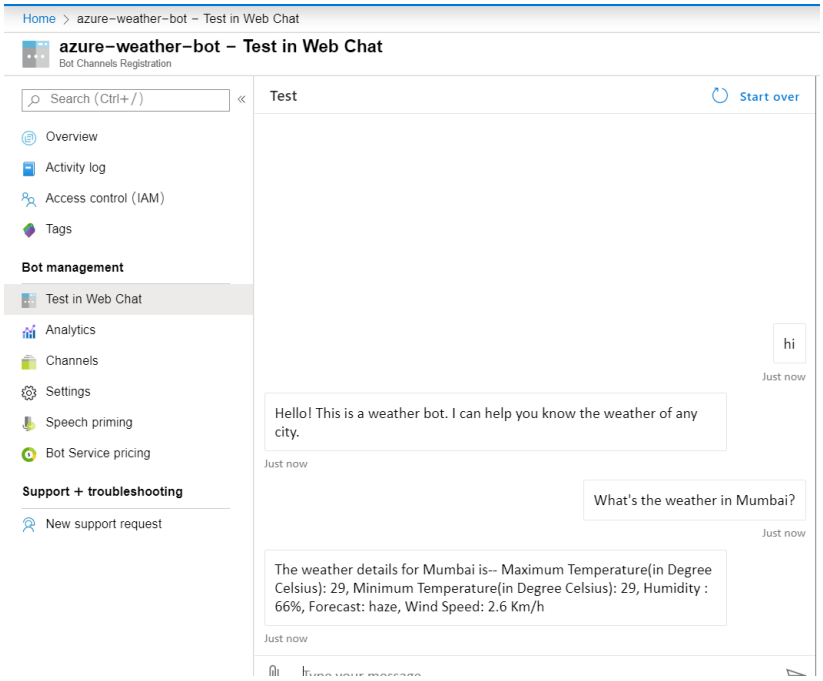
- Run `git init` to initialize an empty git repository
- Git add, commit with an appropriate commit message
- Push the code to the remote repo using `git push <alias> master –f`
- This prompts for a username and password. Go to the 'Deployment Credentials' section and copy the username and password to enter in the prompt.



- Once the credentials are correctly entered, the app deployment to azure is completed.

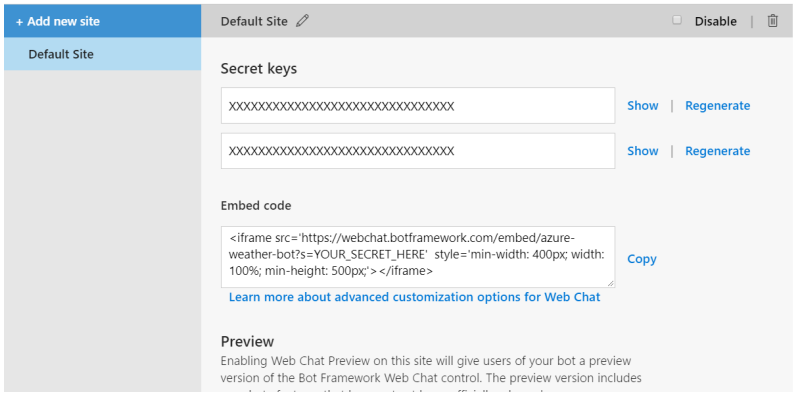
4.4 Create bot channel Registration

- Now in the Azure portal, create a bot channel registration.
- Provide the bot handle, resource group and other fields.
- For Message endpoint, provide: <URL from the web app created above>api/messages. Click Create to create the bot.
- Once your web channel registration gets done, open the bot and then click 'test in web chat.' If the chat works fine, our deployment is a success.



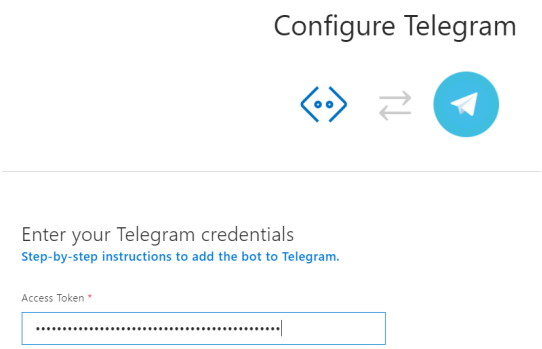
4.5 Deployment

- Go to the channels section of your bot.
- The bot can be deployed as an embedding to an existing HTML page by selecting the get bot embedded code option



4.5.1.1 Telegram Deployment

- Open the telegram app and search for botfather(it is an inbuilt bot used to create other bots)
- Start a conversation with botfather and enter /newbot to create a newbot.
- Give a name to your bot
- Give a username to your bot, which must end in _bot.
- This generates an access token. Enter that access token after clicking the telegram channel in your bot app and click save.



- Now, search the username of the bot in telegram and start conversation with your bot.