# Weather Chatbot using Microsoft Azur Luis Framework

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.

Formulating responses to questions in natural language is one of the most typical Examples of Natural Language Processing applied in various enterprises' end-use applications.

## Business case:

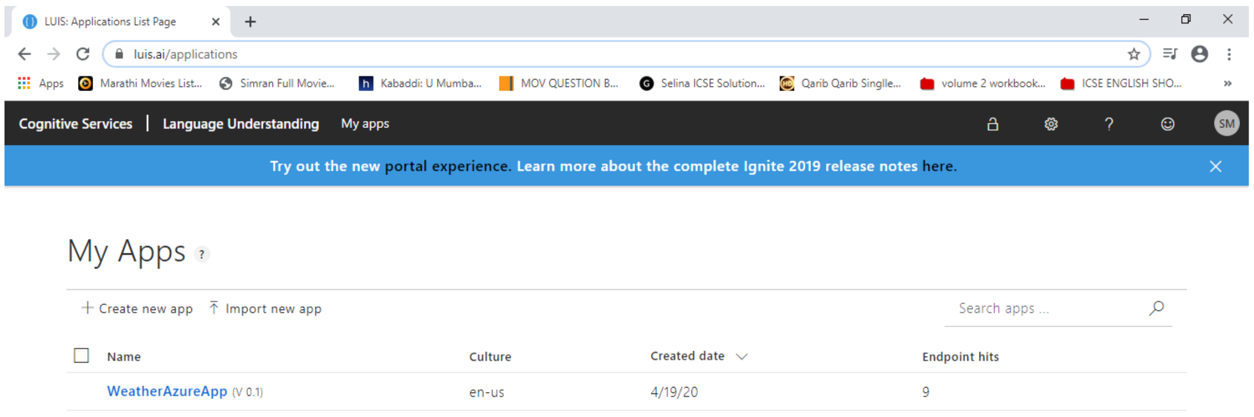
To get the weather information about the city.

Application Flow:

1. User Logs into any channel (eg: Telegram Bot)
2. User queries the weather for any city
3. The telegram bot will match the utterance with the luis intent
4. The control comes to the python application app.py which is running
5. It then calls the weather api to get the weather details
6. The weather details are formed as a response
7. The weather details are sent to luis web app
8. Then it is sent to the requested channel: eg telegram in this case

## Implementation process :

1. Create your account on https://www.luis.ai/
2. Create a new app as below:

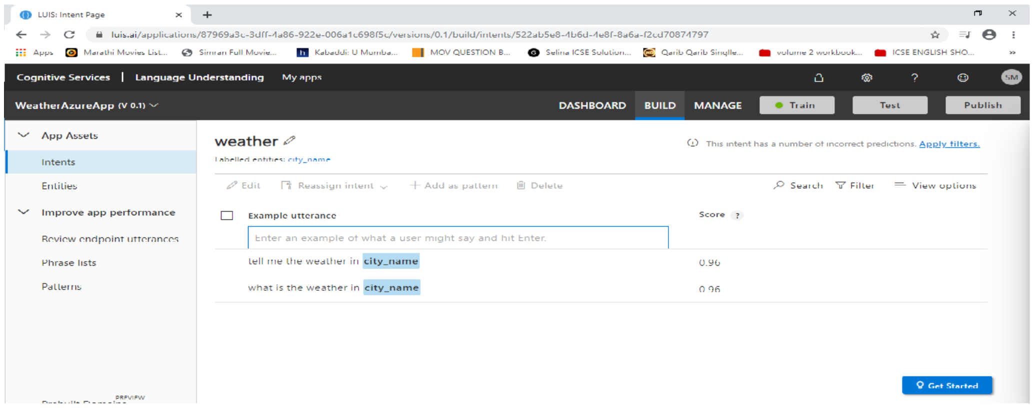


Try out the new portal experience. Learn more about the complete Ignite 2019 release notes here.

My Apps

| Name | Culture | Created date | Endpoint hits |
|---|---|---|---|
| WeatherAzureApp (V 0.1) | en-us | 4/19/20 | 9 |

3. Created an intent as weather and trained with few utterances using the entity as ‘city_name’ of which the weather information is desired by the user


LUIS: Intent Page
luis.ai/applications/87969a3c-3dff-4a86-922e-006a1c698f5c/versions/0.1/build/intents/522ab5e8-4b6d-4e8f-8a6a-f2cd70874797
Cognitive Services | Language Understanding My apps
WeatherAzureApp (V 0.1)
DASHBOARD BUILD MANAGE Train Test Publish
App Assets
Intents
Entities
Improve app performance
Review endpoint utterances
Phrase lists
Patterns
Prebuilt Domains PREVIEW
Edit Reassign intent Add as pattern Delete Search Filter View options
Get Started

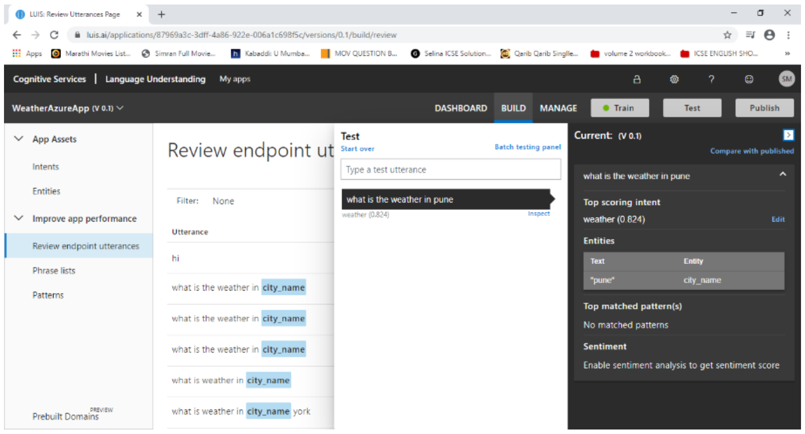
weather

Labelled entities: city_name

This intent has a number of incorrect predictions. Apply filters.

| Example utterance | Score |
|---|---|
| Enter an example of what a user might say and hit Enter. | |
| tell me the weather in city_name | 0.96 |
| what is the weather in city_name | 0.96 |

4. Train and test the model to check that the model is identifying the name of the city as entity ‘city_name’


LUIS: Review Utterances Page
luis.ai/applications/87969a3c-3dff-4a86-922e-006a1c698f5c/versions/0.1/build/review
Cognitive Services | Language Understanding My apps
WeatherAzureApp (V 0.1)
DASHBOARD BUILD MANAGE Train Test Publish
App Assets
Intents
Entities
Improve app performance
Review endpoint utterances
Phrase lists
Patterns
Prebuilt Domains PREVIEW
Test
Start over
Batch testing panel
Type a test utterance
Inspect
Compare with published
Edit

Review endpoint ut

Filter: None

| Utterance |
|---|
| hi |
| what is the weather in city_name |
| what is the weather in city_name |
| what is the weather in city_name |
| what is weather in city_name |
| what is weather in city_name york |

what is the weather in pune
weather (0.824)

Current: (V 0.1)

what is the weather in pune

Top scoring intent
weather (0.824)

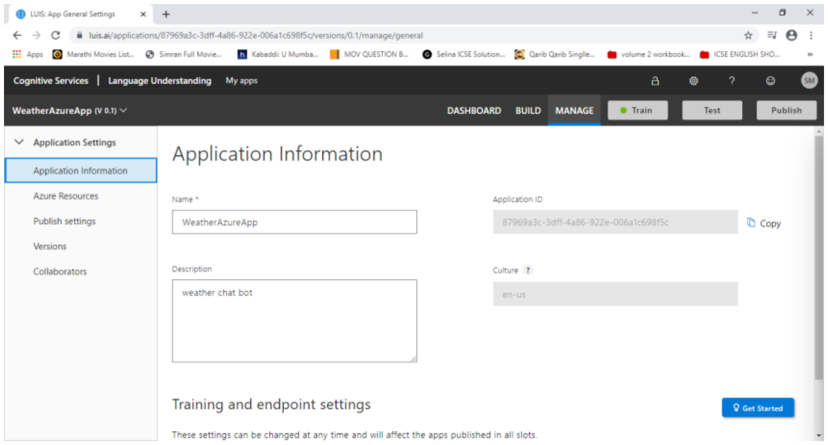
Entities

| Text | Entity |
|---|---|
| "pune" | city_name |

Top matched pattern(s)
No matched patterns

Sentiment
Enable sentiment analysis to get sentiment score

5. Build and publish the app, to get the following details
6. Get the LUIS_APP_ID from the page below:


LUIS: App General Settings
luis.ai/applications/87969a3c-3dff-4a86-922e-006a1c698f5c/versions/0.1/manage/general
Cognitive Services | Language Understanding My apps
WeatherAzureApp (V 0.1)
DASHBOARD BUILD MANAGE Train Test Publish
Application Settings
Application Information
Azure Resources
Publish settings
Versions
Collaborators
Copy
Get Started

Application Information

Name *: WeatherAzureApp

Application ID: 87969a3c-3dff-4a86-922e-006a1c698f5c

Description: weather chat bot

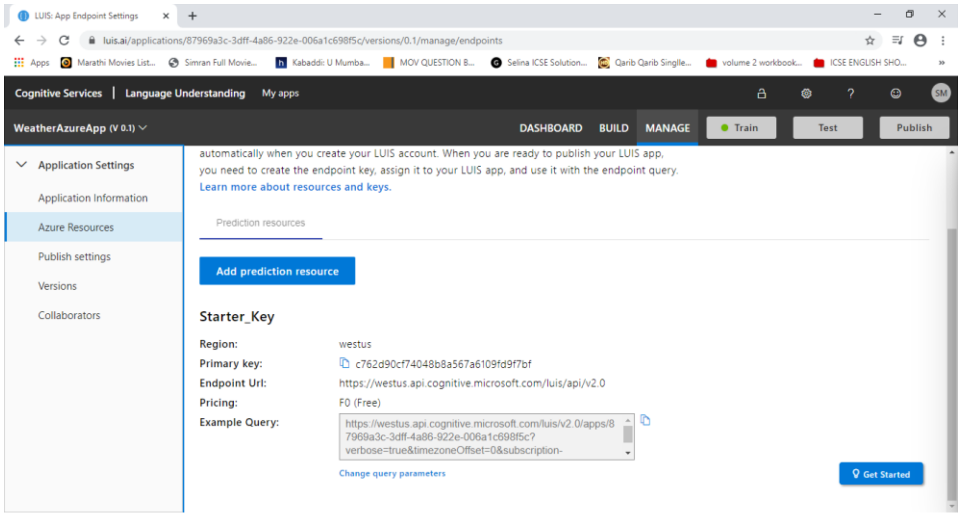
Culture: en-us

Training and endpoint settings

These settings can be changed at any time and will affect the apps published in all slots.

7. Get the other credentials :

8. Primary key and Endpoint url will be used as LUIS_ENDPOINT_KEY and LUIS_ENDPOINT, which will be stroed in our ‘config.ini’ for the python app


LUIS: App Endpoint Settings
luis.ai/applications/87969a3c-3dff-4a86-922e-006a1c698f5c/versions/0.1/manage/endpoints
Cognitive Services | Language Understanding My apps
WeatherAzureApp (V 0.1)
DASHBOARD BUILD MANAGE Train Test Publish
Application Settings
Application Information
Azure Resources
Publish settings
Versions
Collaborators
Prediction resources
Add prediction resource
Change query parameters
Get Started

automatically when you create your LUIS account. When you are ready to publish your LUIS app, you need to create the endpoint key, assign it to your LUIS app, and use it with the endpoint query. Learn more about resources and keys.

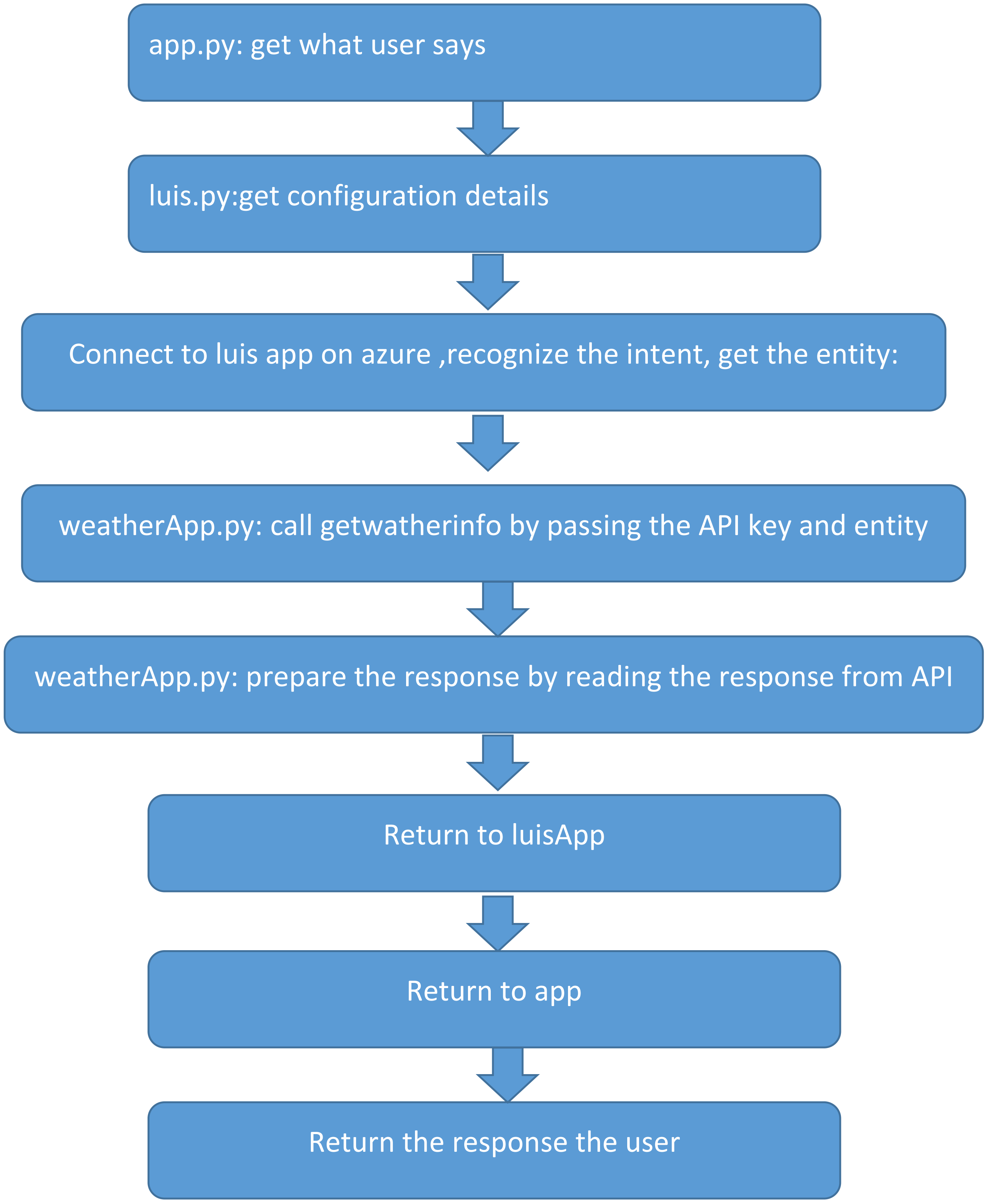
**Starter_Key**

| | |
|---|---|
| Region: | westus |
| Primary key: | c762d90cf74048b8a567a6109fd9f7bf |
| Endpoint Url: | https://westus.api.cognitive.microsoft.com/luis/api/v2.0 |
| Pricing: | F0 (Free) |
| Example Query: | https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/87969a3c-3dff-4a86-922e-006a1c698f5c?verbose=true&timezoneOffset=0&subscription- |

9. Get the weather api from https://home.openweathermap.org/, sign in/signup, and create an API Key for calling the current weather data API.
    a. This api key is used to access the api provided.
    b. “pyowm” module has to be imported after installation.
    c. This has OWM class, which is instantiated in python application, from where the weather_at_place () method is called.

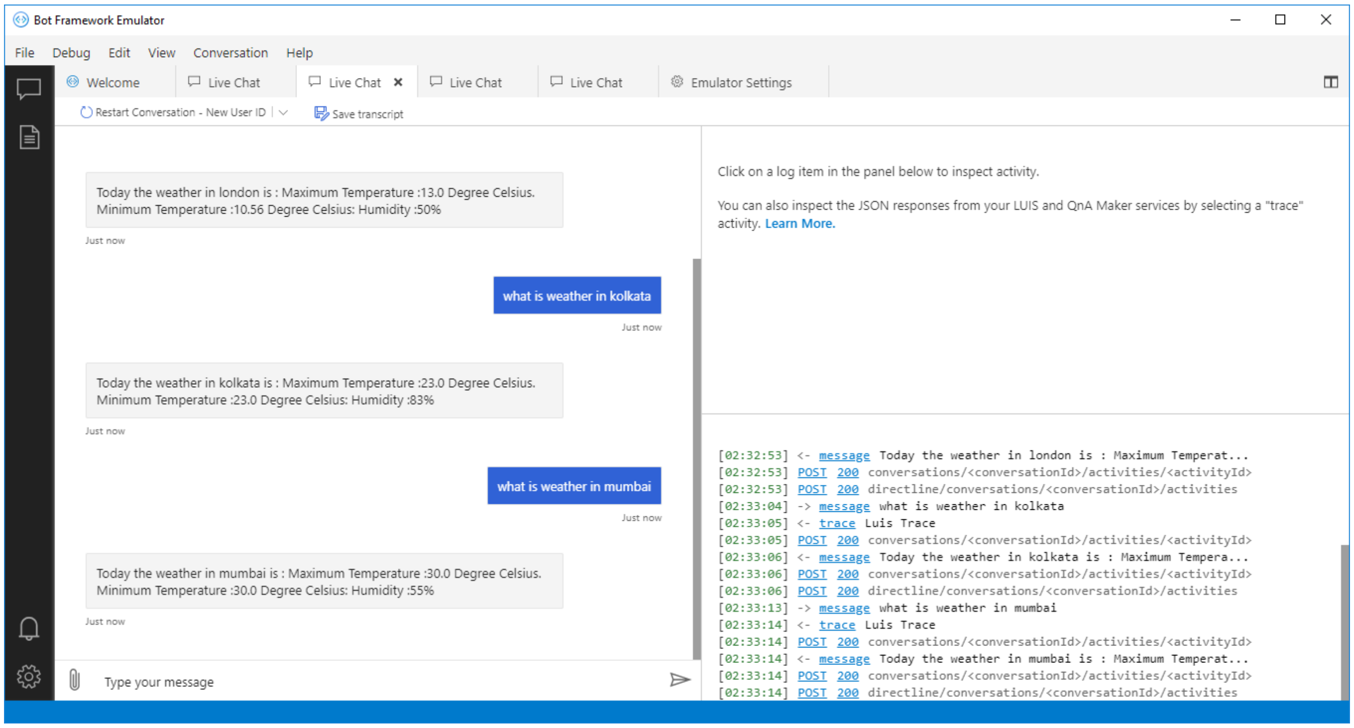
10. Install flask and other dependencies from “https://github.com/Microsoft/BotFramework-Emulator/releases “

## 11. Python Code implementation:


app.py: get what user says
luis.py:get configuration details
Connect to luis app on azure ,recognize the intent, get the entity:
weatherApp.py: call getwatherinfo by passing the API key and entity
weatherApp.py: prepare the response by reading the response from API
Return to luisApp
Return to app
Return the response the user

12. Run the app.py

13. Also start the bot emulator and test your bot



Today the weather in london is : Maximum Temperature :13.0 Degree Celsius. Minimum Temperature :10.56 Degree Celsius: Humidity :50%

what is weather in kolkata

Today the weather in kolkata is : Maximum Temperature :23.0 Degree Celsius. Minimum Temperature :23.0 Degree Celsius: Humidity :83%

what is weather in mumbai

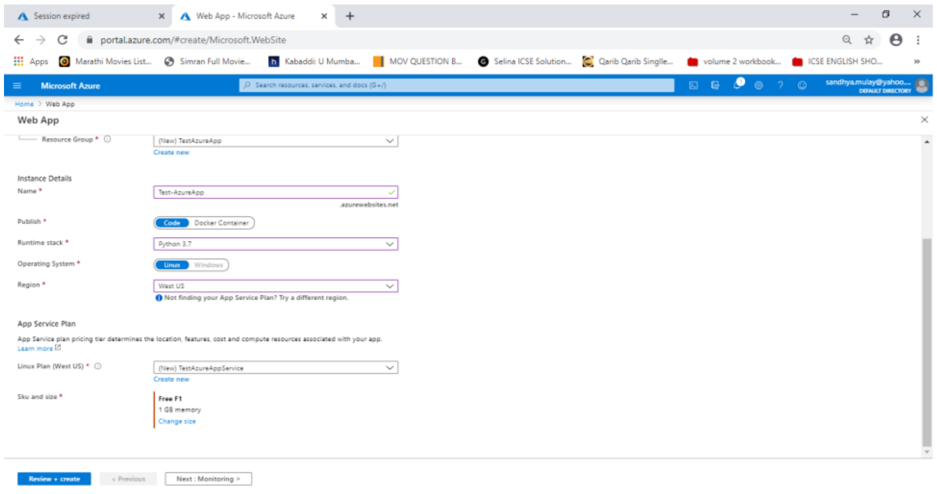
Today the weather in mumbai is : Maximum Temperature :30.0 Degree Celsius. Minimum Temperature :30.0 Degree Celsius: Humidity :55%

Click on a log item in the panel below to inspect activity.

You can also inspect the JSON responses from your LUIS and QnA Maker services by selecting a "trace" activity. Learn More.

```
[02:32:53] <- message Today the weather in london is : Maximum Temperat...
[02:32:53] POST 200 conversations/<conversationId>/activities/<activityId>
[02:32:53] POST 200 directline/conversations/<conversationId>/activities
[02:33:04] -> message what is weather in kolkata
[02:33:05] <- trace Luis Trace
[02:33:05] POST 200 conversations/<conversationId>/activities/<activityId>
[02:33:06] <- message Today the weather in kolkata is : Maximum Tempera...
[02:33:06] POST 200 conversations/<conversationId>/activities/<activityId>
[02:33:06] POST 200 directline/conversations/<conversationId>/activities
[02:33:13] -> message what is weather in mumbai
[02:33:14] <- trace Luis Trace
[02:33:14] POST 200 conversations/<conversationId>/activities/<activityId>
[02:33:14] <- message Today the weather in mumbai is : Maximum Temperat...
[02:33:14] POST 200 conversations/<conversationId>/activities/<activityId>
[02:33:14] POST 200 directline/conversations/<conversationId>/activities
```

14. Now we have to deploy this to azure  by creating a web app
15. Go to azure , create your account
16. Search for Web App and fill in the form as below:



Web App

Resource Group *: (New) TestAzureApp

Instance Details

Name *: Test-AzureApp .azurewebsites.net

Publish *: [✓ Code] Docker Container

Runtime stack *: Python 3.7

Operating System *: [✓ Linux] Windows

Region *: West US

Not finding your App Service Plan? Try a different region.

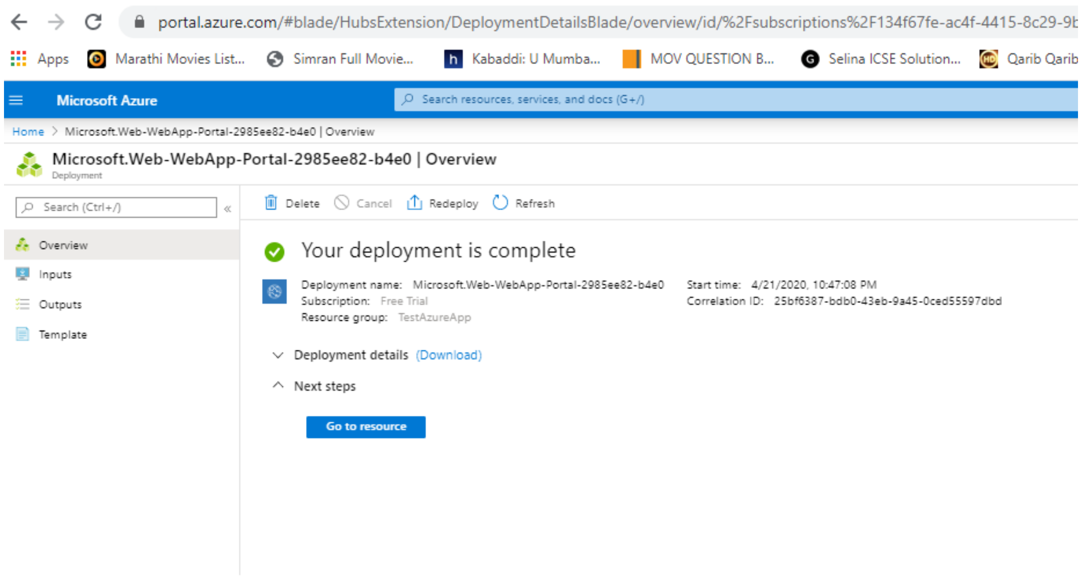
App Service Plan

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

Linux Plan (West US) *: (New) TestAzureAppService

Sku and size *: Free F1, 1 GB memory

17. Hit Review + create ===➔ Create



portal.azure.com/#blade/HubsExtension/DeploymentDetailsBlade/overview/id/%2Fsubscriptions%2F134f67fe-ac4f-4415-8c29-9b

Microsoft.Web-WebApp-Portal-2985ee82-b4e0 | Overview

Your deployment is complete

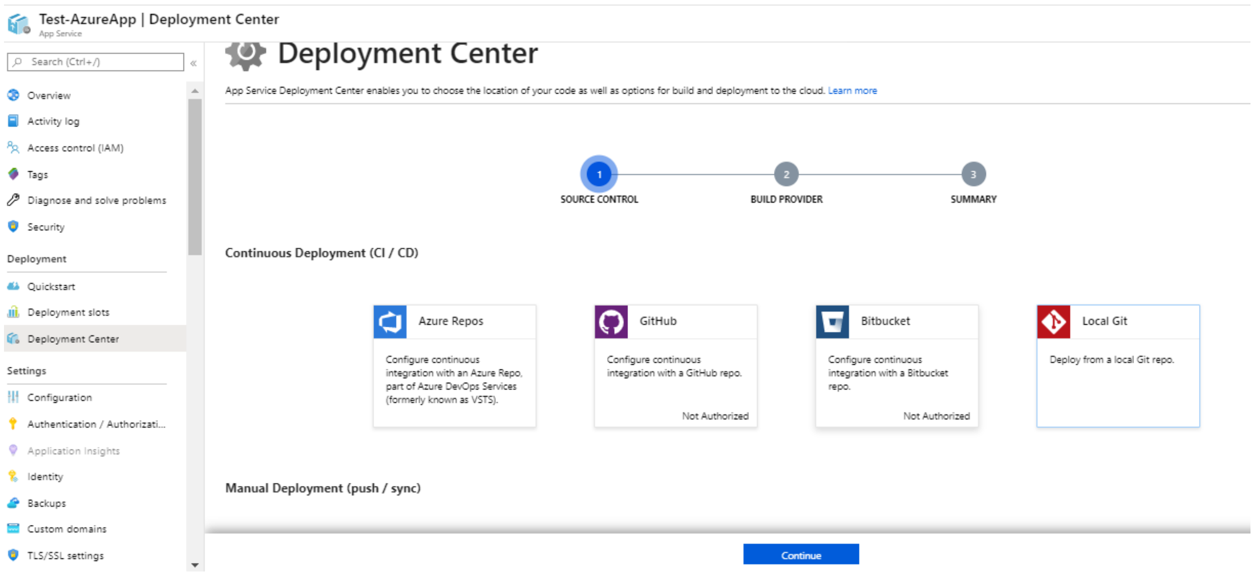
Deployment name: Microsoft.Web-WebApp-Portal-2985ee82-b4e0
Subscription: Free Trial
Resource group: TestAzureApp
Start time: 4/21/2020, 10:47:08 PM
Correlation ID: 25bf6387-bdb0-43eb-9a45-0ced55597dbd

Deployment details (Download)

Next steps

Go to resource

18. Once the deployment is complete , got to deployment centre select ‘Local Git ‘ for source control



Test-AzureApp | Deployment Center

Deployment Center

App Service Deployment Center enables you to choose the location of your code as well as options for build and deployment to the cloud. Learn more

1 SOURCE CONTROL — 2 BUILD PROVIDER — 3 SUMMARY

Continuous Deployment (CI / CD)

Azure Repos: Configure continuous integration with an Azure Repo, part of Azure DevOps Services (formerly known as VSTS).

GitHub: Configure continuous integration with a GitHub repo. Not Authorized

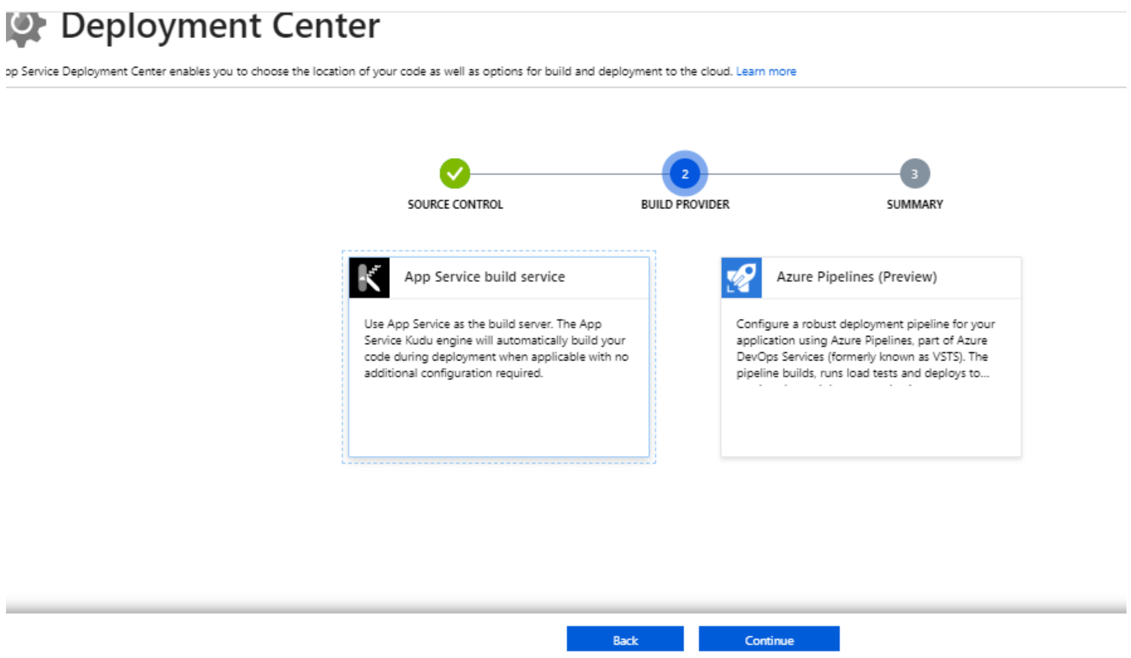
Bitbucket: Configure continuous integration with a Bitbucket repo. Not Authorized

Local Git: Deploy from a local Git repo.

Manual Deployment (push / sync)

Continue

19. Select the kudo ‘App service build provider’ as the build provider and click continue.



# Deployment Center

pp Service Deployment Center enables you to choose the location of your code as well as options for build and deployment to the cloud. Learn more

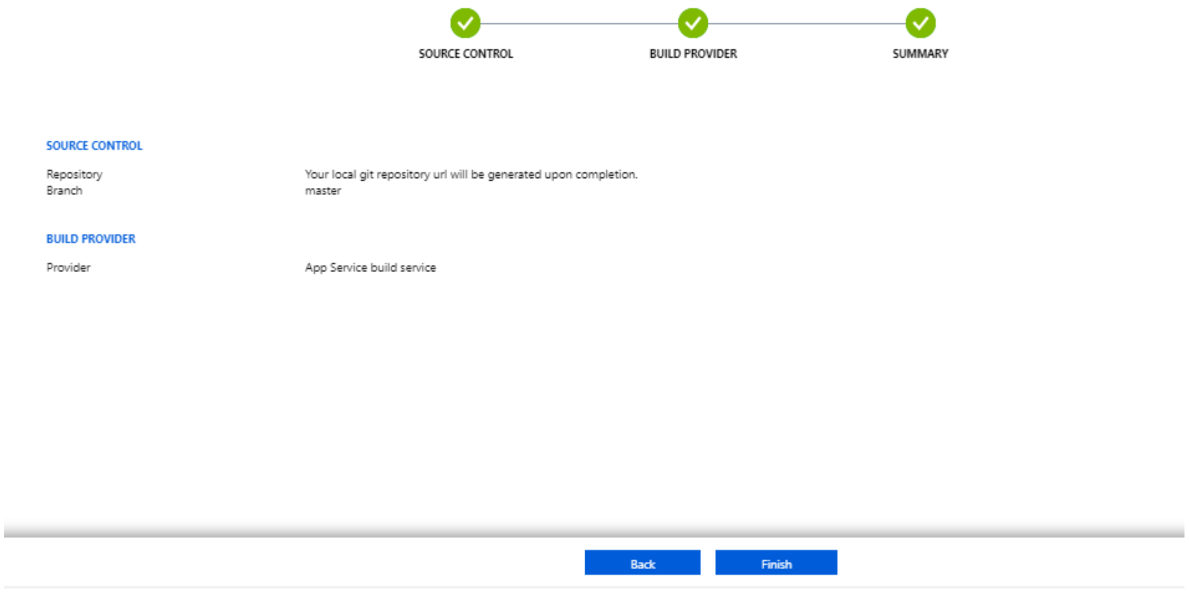
App Service build service

Use App Service as the build server. The App Service Kudu engine will automatically build your code during deployment when applicable with no additional configuration required.

Azure Pipelines (Preview)

Configure a robust deployment pipeline for your application using Azure Pipelines, part of Azure DevOps Services (formerly known as VSTS). The pipeline builds, runs load tests and deploys to...

20. click’finish’ to complete the setup.



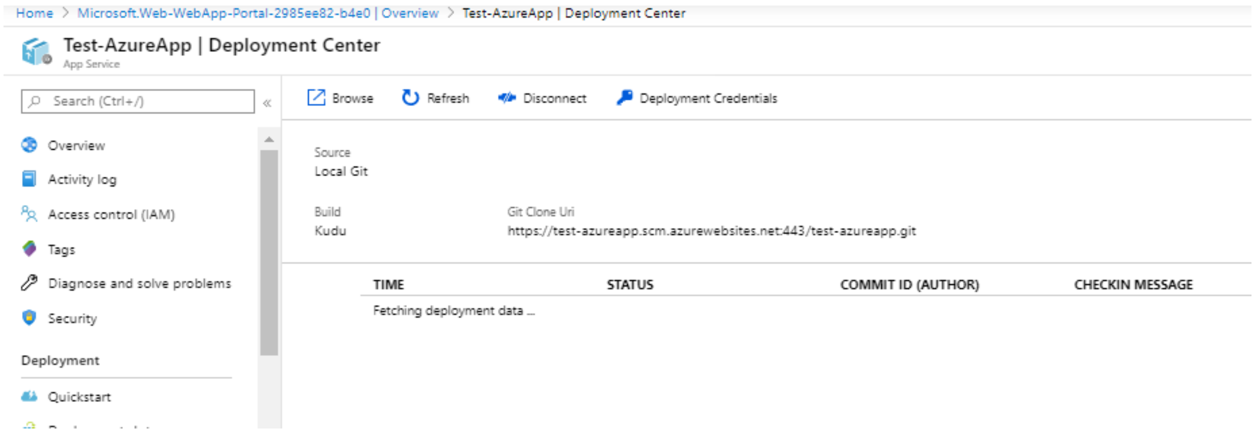
SOURCE CONTROL

| | |
|---|---|
| Repository | Your local git repository url will be generated upon completion. |
| Branch | master |

BUILD PROVIDER

| | |
|---|---|
| Provider | App Service build service |

21. set up is complete .


Home > Microsoft.Web-WebApp-Portal-2985ee82-b4e0 | Overview > Test-AzureApp | Deployment Center

Test-AzureApp | Deployment Center
App Service

Search (Ctrl+/)

Browse · Refresh · Disconnect · Deployment Credentials

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Deployment
Quickstart

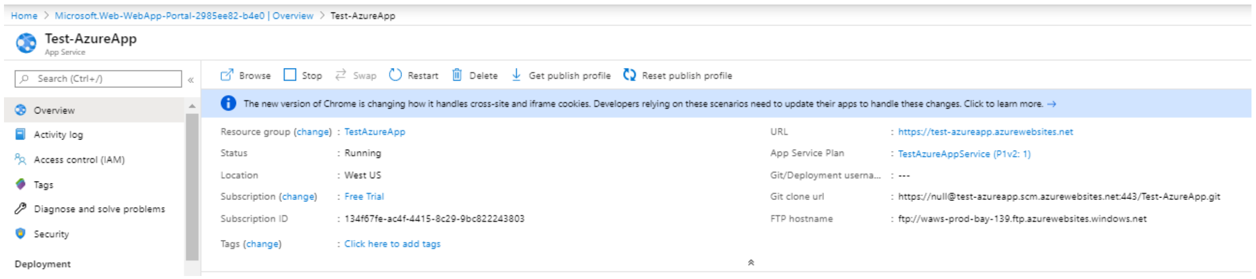
Source
Local Git

Build
Kudu

Git Clone Uri
https://test-azureapp.scm.azurewebsites.net:443/test-azureapp.git

| TIME | STATUS | COMMIT ID (AUTHOR) | CHECKIN MESSAGE |
|---|---|---|---|
| Fetching deployment data ... | | | |

22. check the overview section for the git link


Home > Microsoft.Web-WebApp-Portal-2985ee82-b4e0 | Overview > Test-AzureApp

Test-AzureApp
App Service

Search (Ctrl+/)

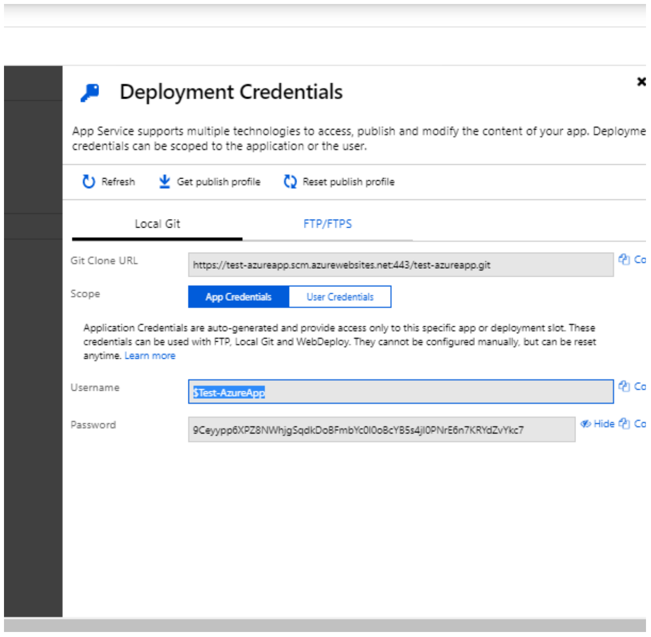
Browse · Stop · Swap · Restart · Delete · Get publish profile · Reset publish profile

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Deployment

The new version of Chrome is changing how it handles cross-site and iframe cookies. Developers relying on these scenarios need to update their apps to handle these changes. Click to learn more. →

| | | | |
|---|---|---|---|
| Resource group (change) | : TestAzureApp | URL | : https://test-azureapp.azurewebsites.net |
| Status | : Running | App Service Plan | : TestAzureAppService (P1v2: 1) |
| Location | : West US | Git/Deployment userna... | : --- |
| Subscription (change) | : Free Trial | Git clone url | : https://null@test-azureapp.scm.azurewebsites.net:443/Test-AzureApp.git |
| Subscription ID | : 134f67fe-ac4f-4415-8c29-9bc822243803 | FTP hostname | : ftp://waws-prod-bay-139.ftp.azurewebsites.windows.net |
| Tags (change) | : Click here to add tags | | |

23. Get the deployment credentials


Deployment Credentials

App Service supports multiple technologies to access, publish and modify the content of your app. Deployme
credentials can be scoped to the application or the user.

Refresh · Get publish profile · Reset publish profile

Local Git · FTP/FTPS

Git Clone URL: https://test-azureapp.scm.azurewebsites.net:443/test-azureapp.git

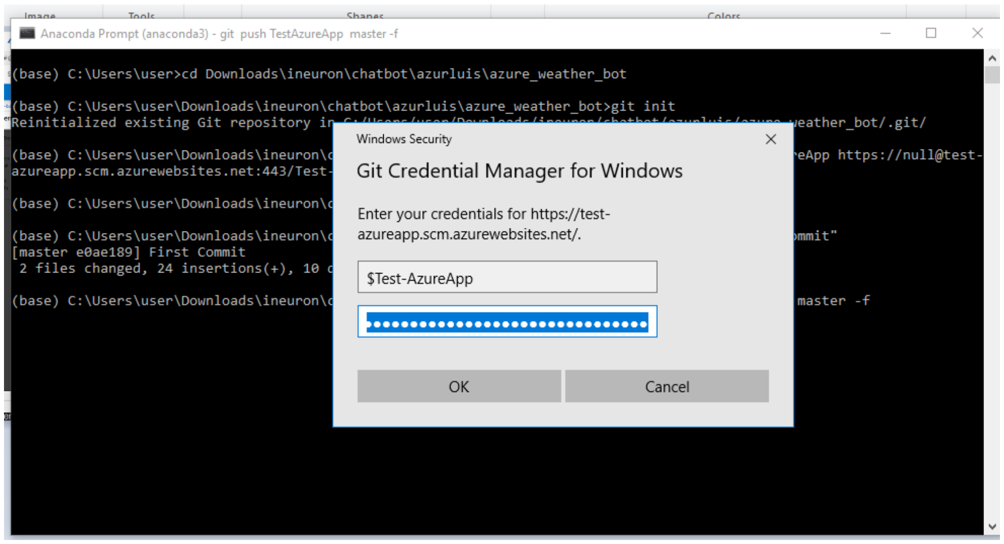
Scope: [✓ App Credentials] User Credentials

Application Credentials are auto-generated and provide access only to this specific app or deployment slot. These credentials can be used with FTP, Local Git and WebDeploy. They cannot be configured manually, but can be reset anytime. Learn more

Username: $Test-AzureApp

Password: 9Ceyypp6XPZ8NWhjgSqdkDoBFmbYc0l0oBcYB5s4jl0PNrE6n7KRYdZvYkc7

24. go to project folder on the command prompt
    a. git init
    b. git remote add TestAzureApp [https://null@test-azureapp.scm.azurewebsites.net:443/Test-AzureApp.git](https://null@test-azureapp.scm.azurewebsites.net:443/Test-AzureApp.git)
    c. git add .
    d. gir commit –m ‘First Commit’
    e. git push TestAzureApp master –f
    f. add the credentials taken from step 23



Anaconda Prompt (anaconda3) - git push TestAzureApp master -f

```
(base) C:\Users\user>cd Downloads\ineuron\chatbot\azurluis\azure_weather_bot

(base) C:\Users\user\Downloads\ineuron\chatbot\azurluis\azure_weather_bot>git init
Reinitialized existing Git repository in C:/Users/user/Downloads/ineuron/chatbot/azurluis/azure_weather_bot/.git/

(base) C:\Users\user\Downloads\ineuron\c                                              eApp https://null@test-
azureapp.scm.azurewebsites.net:443/Test-

(base) C:\Users\user\Downloads\ineuron\c

(base) C:\Users\user\Downloads\ineuron\c                                              mmit"
[master e0ae189] First Commit
 2 files changed, 24 insertions(+), 10 c

(base) C:\Users\user\Downloads\ineuron\c                                              master -f
```

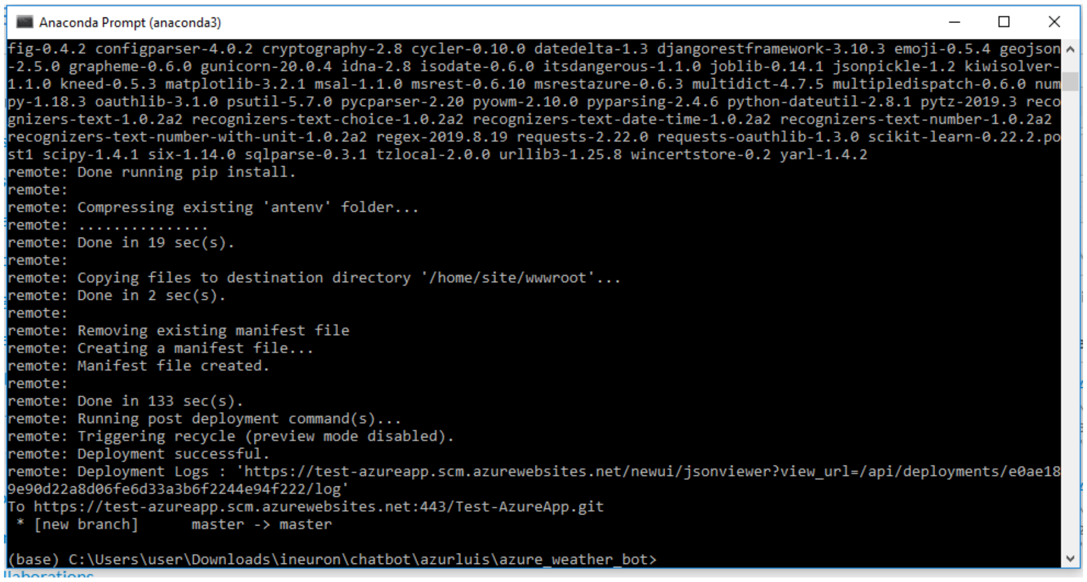
Windows Security

Git Credential Manager for Windows

Enter your credentials for https://test-azureapp.scm.azurewebsites.net/.

$Test-AzureApp

••••••••••••••••••••••••••••••••

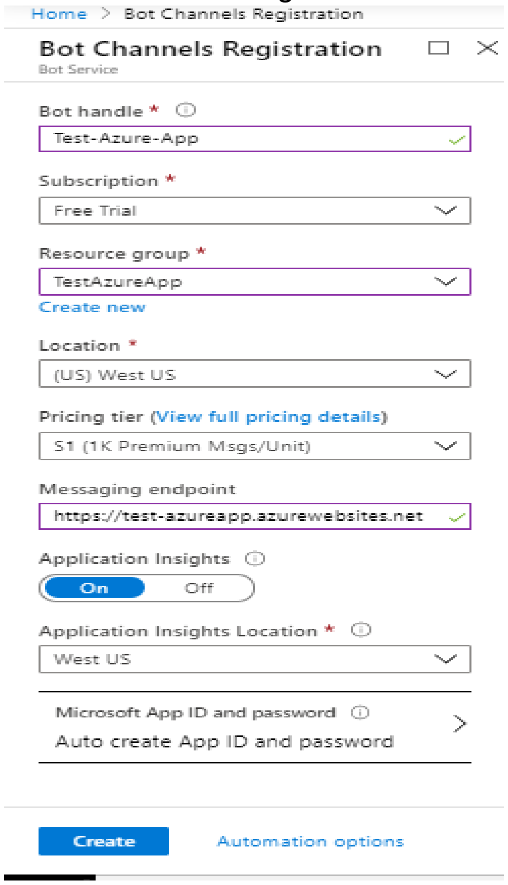


Anaconda Prompt (anaconda3)

```
fig-0.4.2 configparser-4.0.2 cryptography-2.8 cycler-0.10.0 datedelta-1.3 djangorestframework-3.10.3 emoji-0.5.4 geojson
-2.5.0 graphene-0.6.0 gunicorn-20.0.4 idna-2.8 isodate-0.6.0 itsdangerous-1.1.0 joblib-0.14.1 jsonpickle-1.2 kiwisolver-
1.1.0 kneed-0.5.3 matplotlib-3.2.1 msal-1.1.0 msrest-0.6.10 msrestazure-0.6.3 multidict-4.7.5 multipledispatch-0.6.0 num
py-1.18.3 oauthlib-3.1.0 psutil-5.7.0 pycparser-2.20 pyowm-2.10.0 pyparsing-2.4.6 python-dateutil-2.8.1 pytz-2019.3 reco
gnizers-text-1.0.2a2 recognizers-text-choice-1.0.2a2 recognizers-text-date-time-1.0.2a2 recognizers-text-number-1.0.2a2
recognizers-text-number-with-unit-1.0.2a2 regex-2019.8.19 requests-2.22.0 requests-oauthlib-1.3.0 scikit-learn-0.22.2.po
st1 scipy-1.4.1 six-1.14.0 sqlparse-0.3.1 tzlocal-2.0.0 urllib3-1.25.8 wincertstore-0.2 yarl-1.4.2
remote: Done running pip install.
remote:
remote: Compressing existing 'antenv' folder...
remote: ...............
remote: Done in 19 sec(s).
remote:
remote: Copying files to destination directory '/home/site/wwwroot'...
remote: Done in 2 sec(s).
remote:
remote: Removing existing manifest file
remote: Creating a manifest file...
remote: Manifest file created.
remote:
remote: Done in 133 sec(s).
remote: Running post deployment command(s)...
remote: Triggering recycle (preview mode disabled).
remote: Deployment successful.
remote: Deployment Logs : 'https://test-azureapp.scm.azurewebsites.net/newui/jsonviewer?view_url=/api/deployments/e0ae18
9e90d22a8d06fe6d33a3b6f2244e94f222/log'
To https://test-azureapp.scm.azurewebsites.net:443/Test-AzureApp.git
 * [new branch]      master -> master

(base) C:\Users\user\Downloads\ineuron\chatbot\azurluis\azure_weather_bot>
```

25: Bot channel Registration:


Home > Bot Channels Registration

Bot Channels Registration
Bot Service

Bot handle *
Test-Azure-App

Subscription *
Free Trial

Resource group *
TestAzureApp
Create new

Location *
(US) West US

Pricing tier (View full pricing details)
S1 (1K Premium Msgs/Unit)

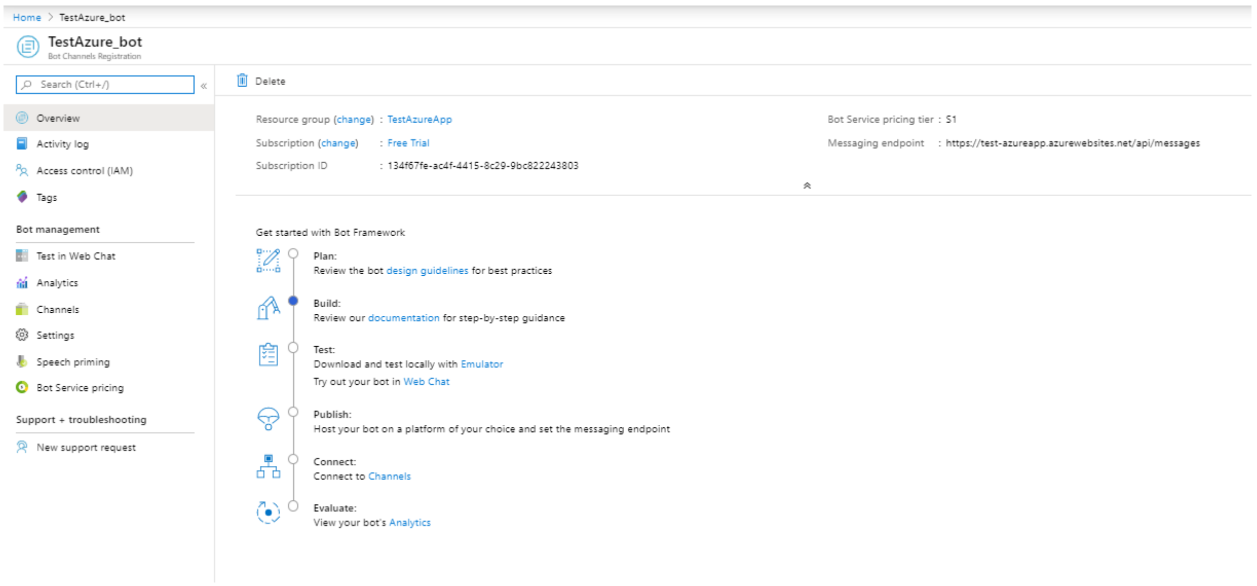
Messaging endpoint
https://test-azureapp.azurewebsites.net

Application Insights
[✓ On] Off

Application Insights Location *
West US

Microsoft App ID and password
Auto create App ID and password

Create    Automation options


Home > TestAzure_bot

TestAzure_bot
Bot Channels Registration

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags

Bot management

- Test in Web Chat
- Analytics
- Channels
- Settings
- Speech priming
- Bot Service pricing

Support + troubleshooting

- New support request

Delete

| | | | |
|---|---|---|---|
| Resource group (change) | : TestAzureApp | Bot Service pricing tier | : S1 |
| Subscription (change) | : Free Trial | Messaging endpoint | : https://test-azureapp.azurewebsites.net/api/messages |
| Subscription ID | : 134f67fe-ac4f-4415-8c29-9bc822243803 | | |

Get started with Bot Framework

Plan:
Review the bot design guidelines for best practices

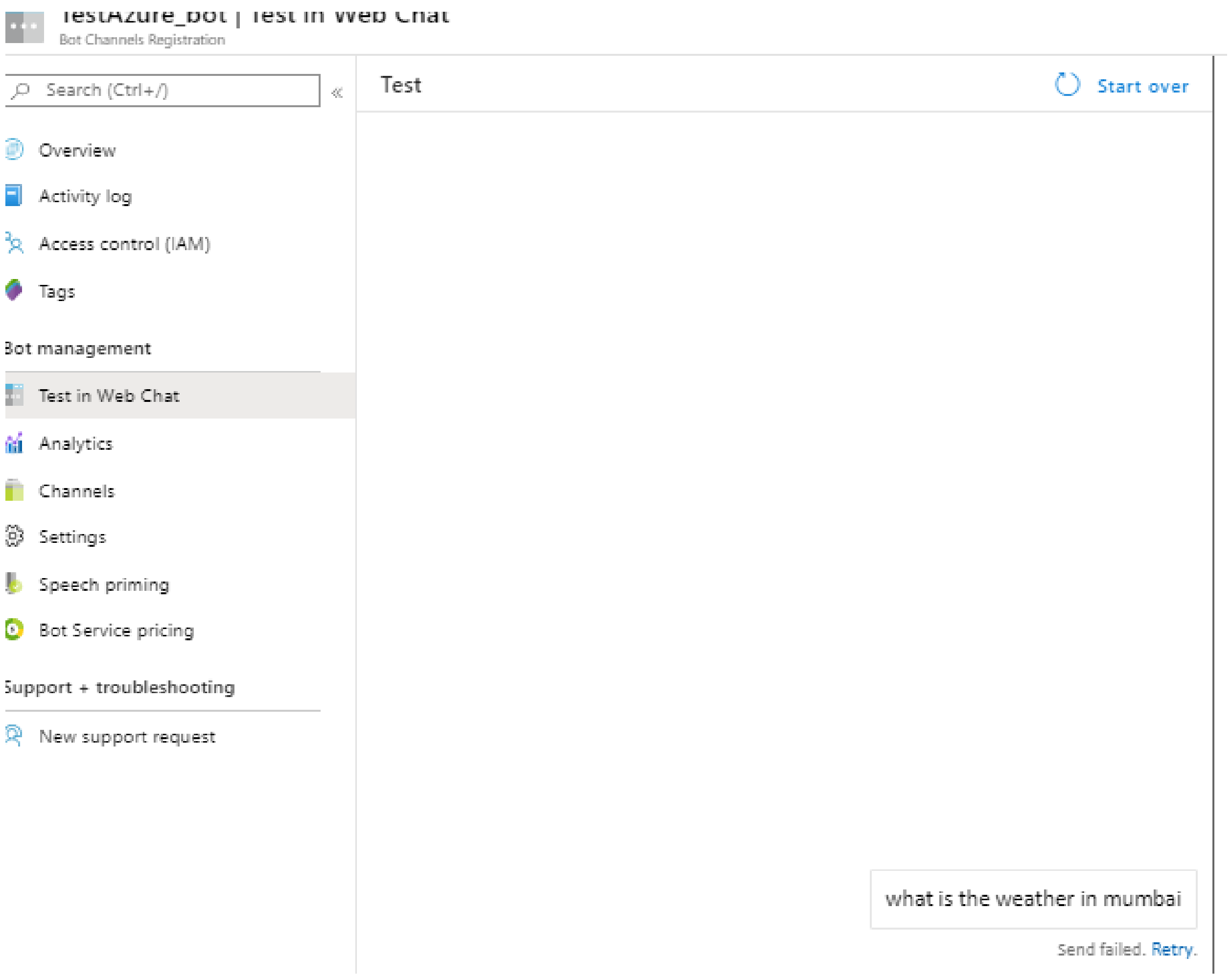
Build:
Review our documentation for step-by-step guidance

Test:
Download and test locally with Emulator
Try out your bot in Web Chat

Publish:
Host your bot on a platform of your choice and set the messaging endpoint

Connect:
Connect to Channels

Evaluate:
View your bot's Analytics



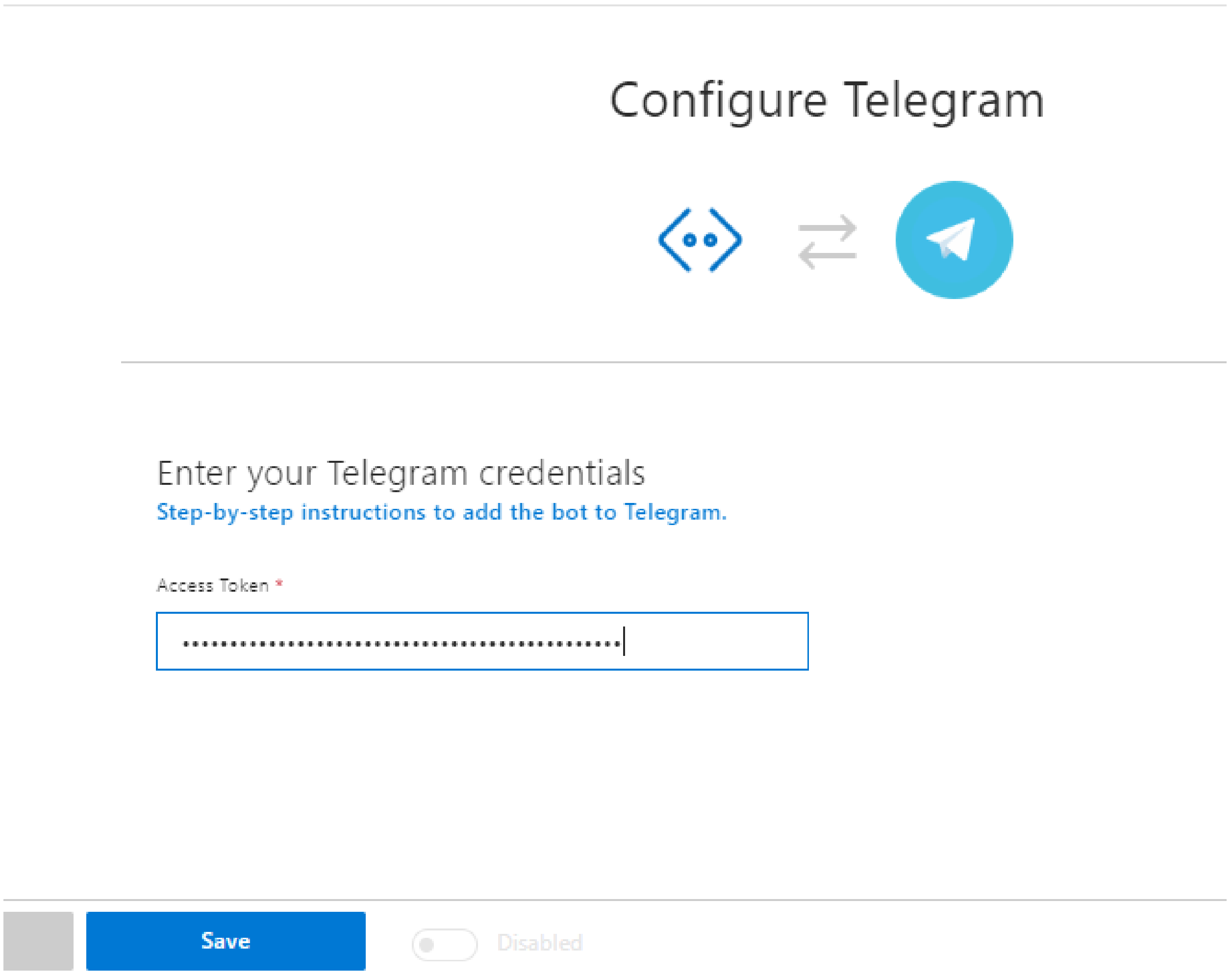
what is the weather in mumbai

Send failed. Retry.

## Telegram integration:

Created a new bot using botfather  /newbot “azureweatherbot”

Copied the access token and used as below



Configure Telegram

Enter your Telegram credentials

Step-by-step instructions to add the bot to Telegram.

Access Token *

••••••••••••••••••••••••••••••••••••••••••••••