**Report: Using cloud technology to develop chatbot**

**Group member**:

Fan Junliang 21448973

Song Ningning 21444579

**Job division**:

Fan Junliang: Reading/writing a TV show review

Whole cloud deploy coding part

Song Ningning: Sharing a hiking route and photos to other users.

Report and presentation

**Chatbot ID**: fjlskr\_bot

**Github project**: https://github.com/fjlskrskr/COMP7940\_groupchatbot.git

**Summary of the app**:

**Functions:**

* Posting topics and adding comments similar to the forum, which can be started and viewed by typing /review;
* Publishing photos and text, which can also be viewed by typing /start. The whole publishing process will jump automatically without command instruction

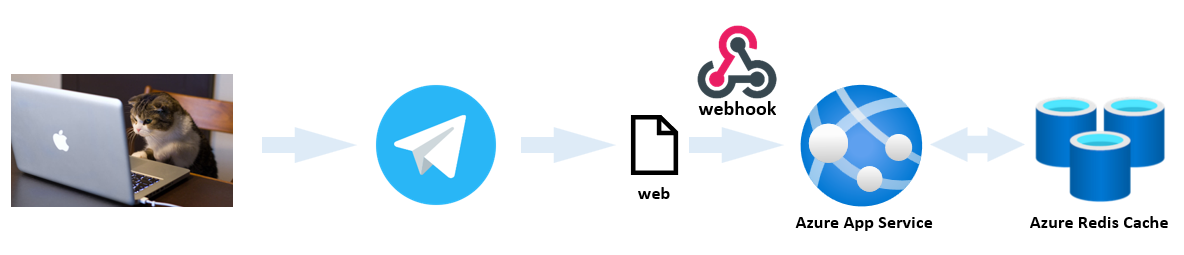
The **database** system we use is Redis. Because our function stores data as text and picture IDs, and the actual data storage capacity is very small, we choose the Azure cache for Redis provided by Azure, which has higher performance than ordinary databases, and supports select scale to view or change the pricing tier for cache.

App host on Azure **cloud platform** (Azure app service)

**Security**: we put all Environmental parameters in the service of App Service, ensure that developers do not get the password of the project during their participation in the project. And Azure will encrypt all data written to the Azure storage account.

One of the reasons for choosing azure is that Azure has a variety of data centers around the world. We can choose areas close to the location to improve performance and reduce the delay in network requests. App Service not only adds the power of Microsoft Azure to our application, such as security, load balancing, autoscaling, and automated management. We can also take advantage of its DevOps capabilities, such as continuous deployment from GitHub, Docker Hub, and other sources. And because Azure App Service is an HTTP-based service for hosting web applications, Therefore, the data transmission path will be changed from directly to the cloud platform by telegram to a web, and the webhook will be automatically transferred to the app service.

The data transmission process is as follows:

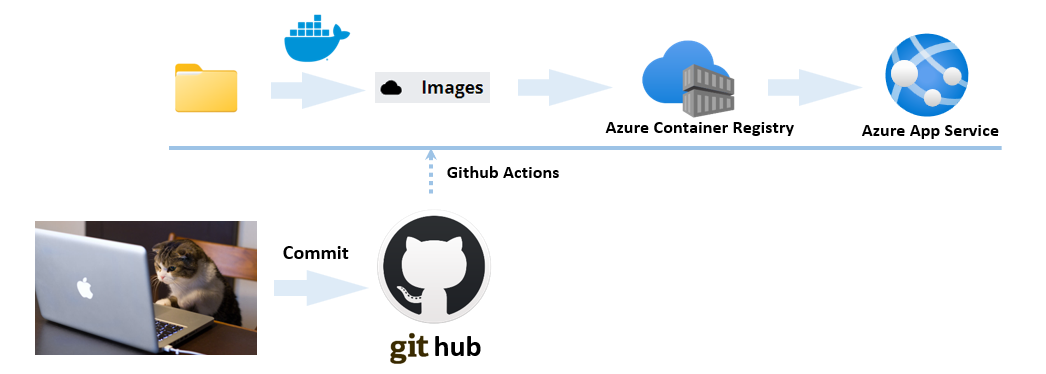


Therefore, App Service will perform elastic and dynamic scaling of cloud computing resources according to the application load, and distribute network traffic among multiple servers to ensure that a single server will not bear too much load.

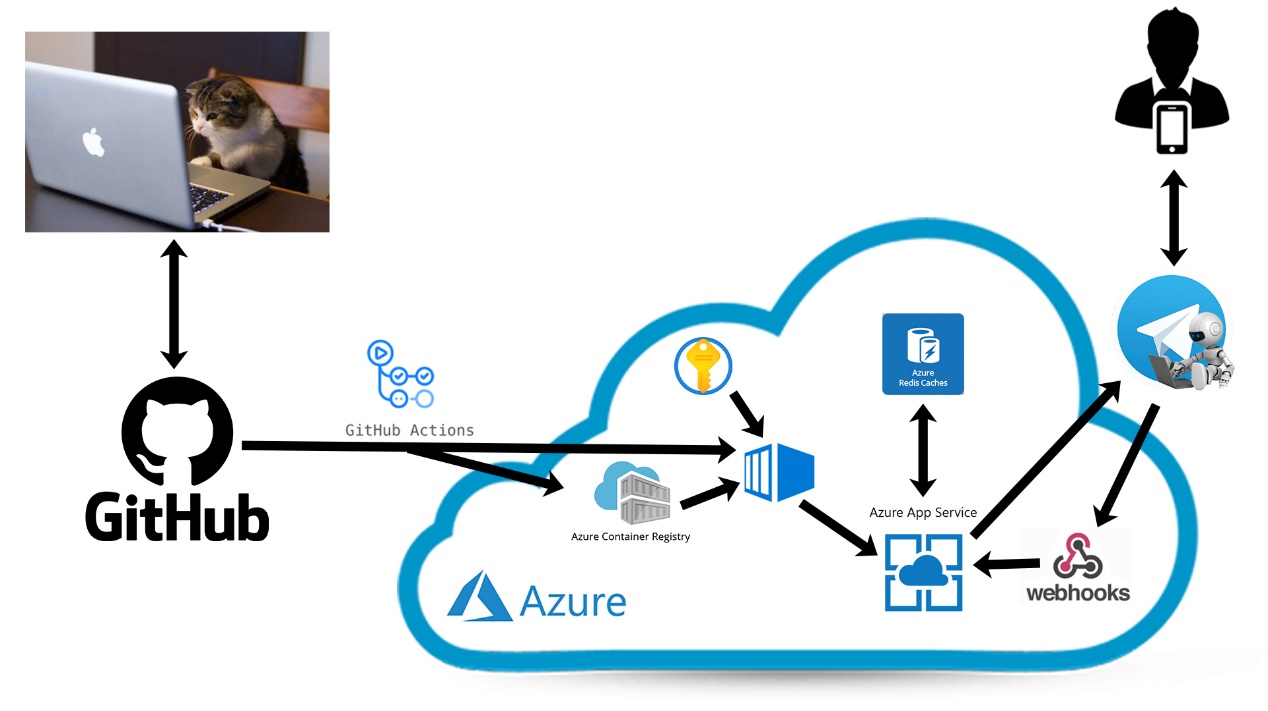
load balancing:

**Workflow**

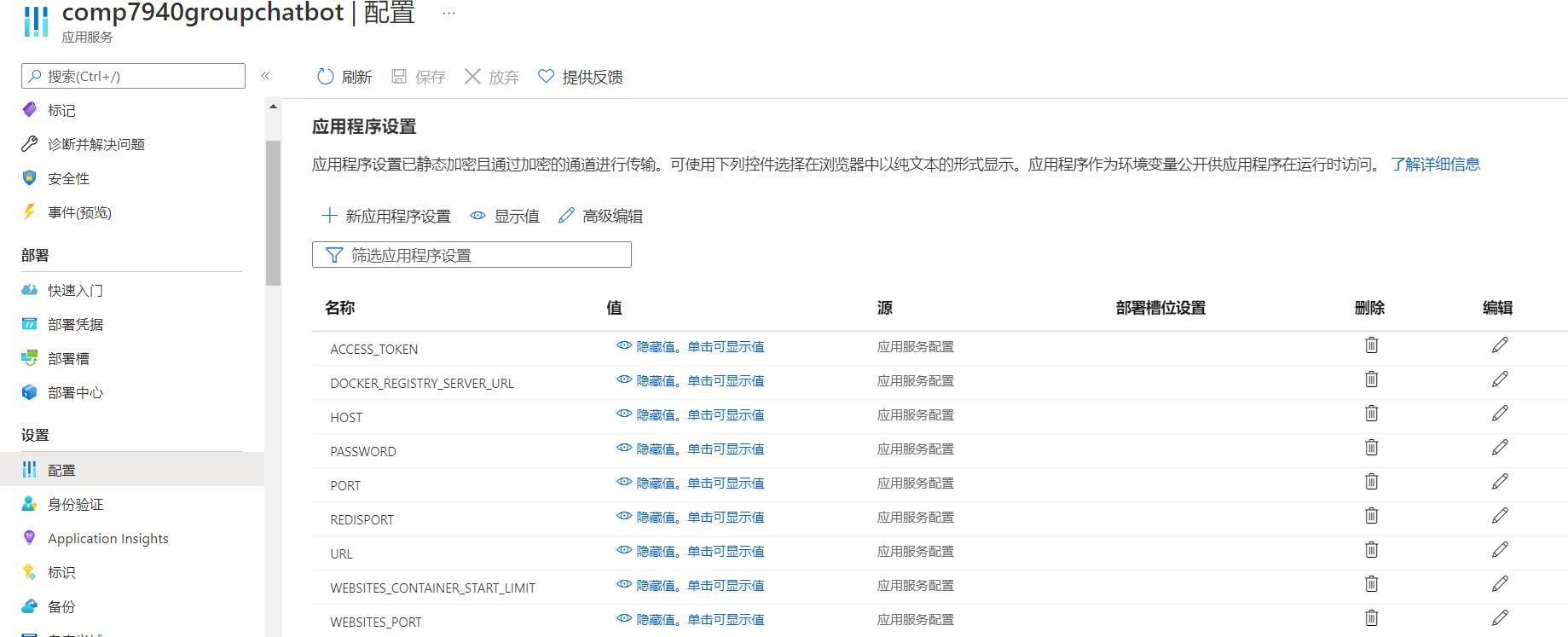
At the same time, we use the GitHub Actions function of GitHub to automatically package the local project into an image when the commit statement is executed, and then push the image to Azure Container Registration, and then the Azure App Service will automatically obtain and deploy the image from Azure Container Registration.

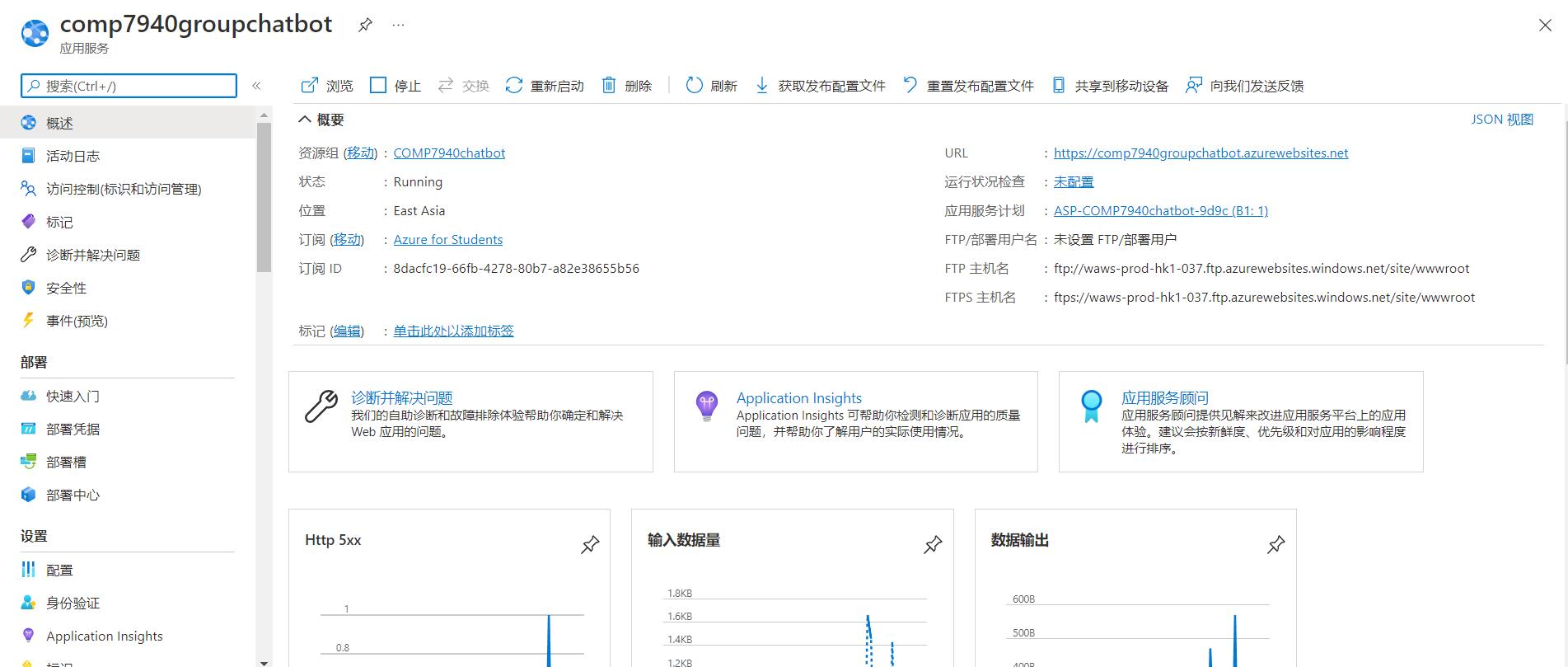


**The summary process is as follows:**

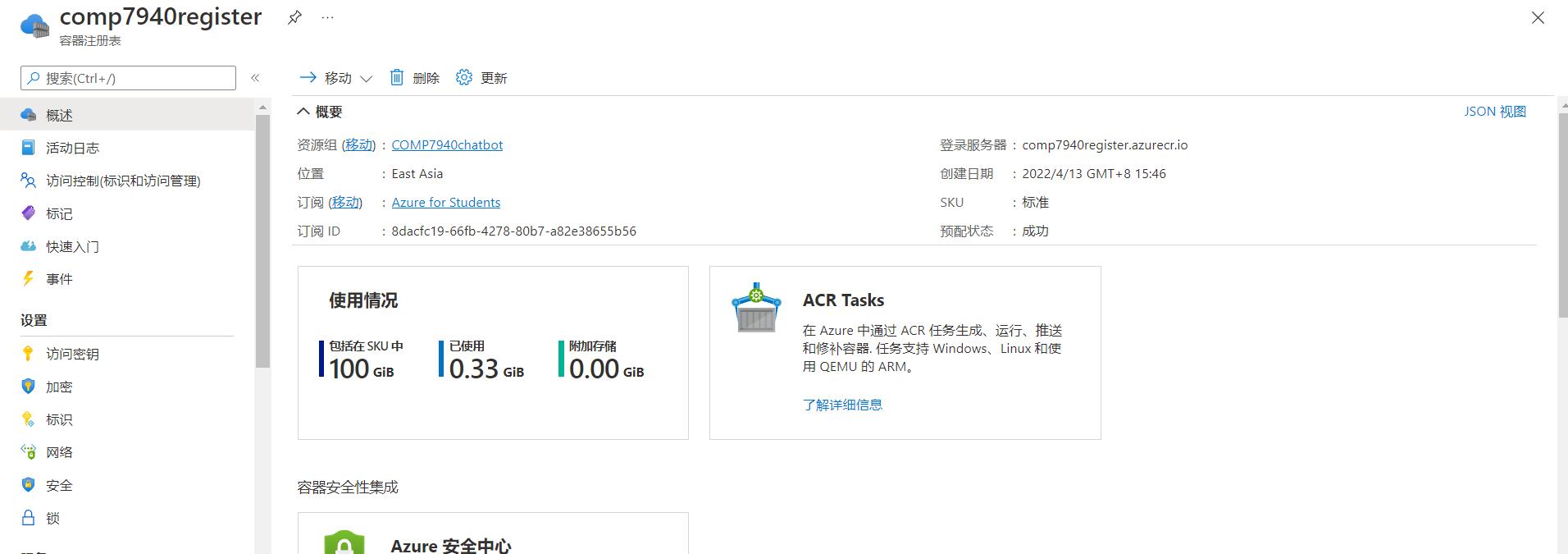


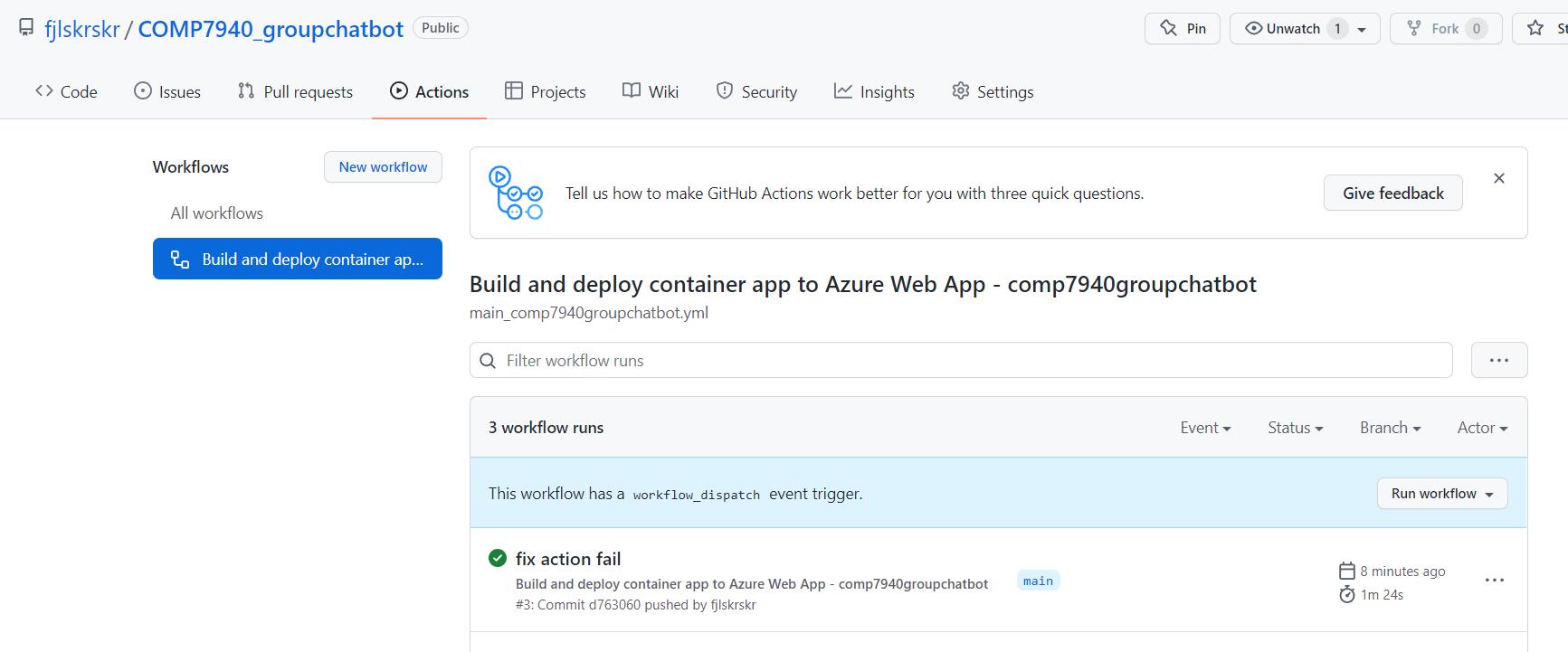
**In order to verify the authenticity of the above cloud service functions used by our team, the following screenshots are attached:**











**References:**

https://blog.csdn.net/u013396714/article/details/121233805

https://www.runoob.com/w3cnote/python-redis-intro.html

https://blog.csdn.net/weixin\_33119123/article/details/112189136

https://github.com/python-telegram-bot/python-telegram-bot/blob/master/examples/README.md

https://www.cnblogs.com/kainhuck/p/13576012.html

https://python-telegram-bot.readthedocs.io/en/latest/telegram.user.html?highlight=telegram.user.User