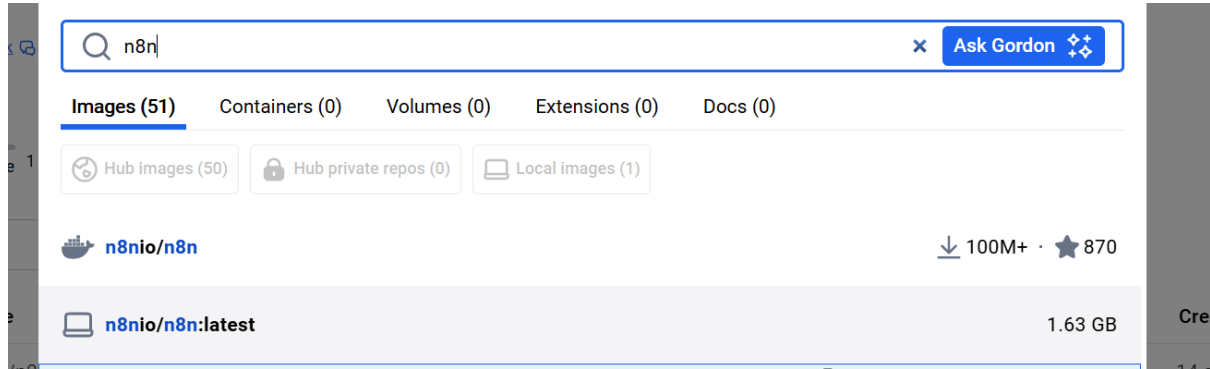


## Create Agentic Workflow in N8N framework:

### Step1: Download docker desktop

<https://docs.docker.com/desktop/setup/install/windows-install/>

### Step2: Open application, and pull N8N image



Step3: run below command to run the container or instance of image:

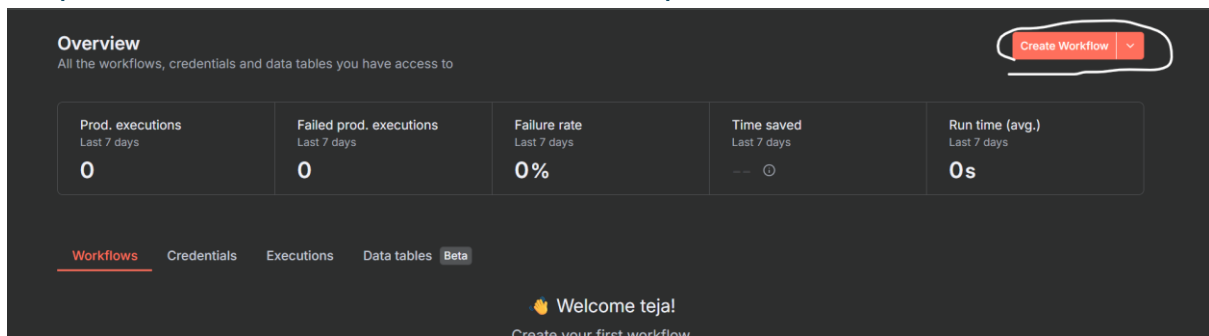
➤ `docker run -it --rm -p 3000:3000 --name n8n n8nio/n8n`

Step4: open in browser localhost:3000

Enter details and next:

A screenshot of the N8N 'Set up owner account' form. The form is titled 'Set up owner account' and has the N8N logo at the top. It contains four input fields: 'Email', 'First Name', 'Last Name', and 'Password'. Below the 'Password' field, there is a note: '8+ characters, at least 1 number and 1 capital letter'. At the bottom of the form, there is a checkbox labeled 'I want to receive security and product updates' and a 'Next' button.

## Step 5: Click create workflow and start the process



## Step 6: or you can import the existing json/workflow



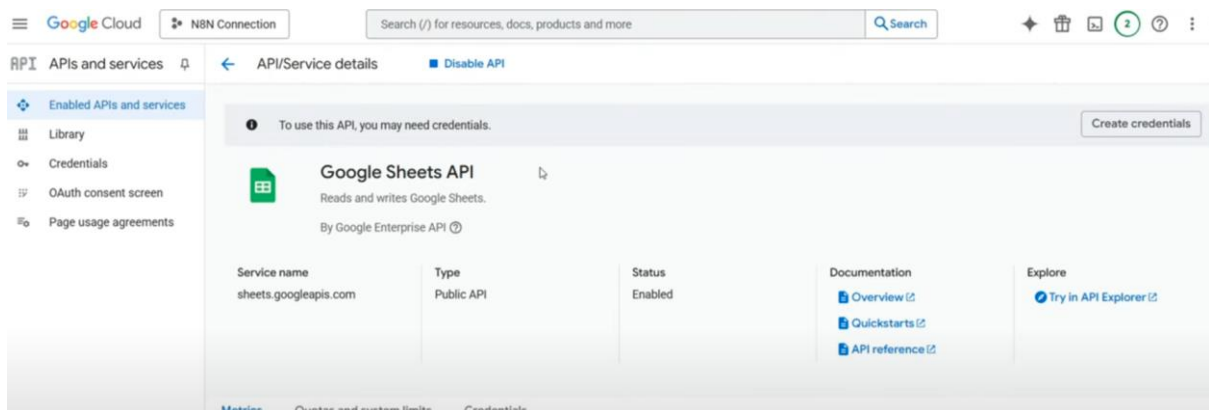
mail-read-workflow.json

## Step 7: Integration with Google API's

7.1: open website - console.cloud.google.com

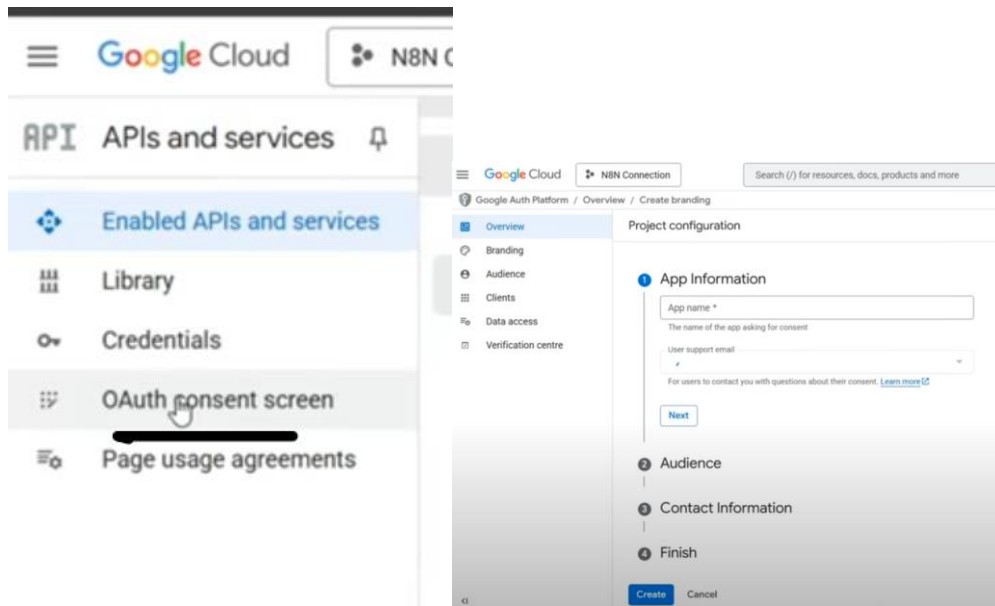
Create and select project

API & Services > Enable APIs & Services > select your app > Enable



7.2: Now generate client and secret:

OAuth consent screen > Enter details and save > Audience (if gmail internal else external) > enter your email and continue > create



Now click on Clients (to create auth id) > add authorized redirect url's – the callback url > save your client id and secret

Step8: Integrate with telegram

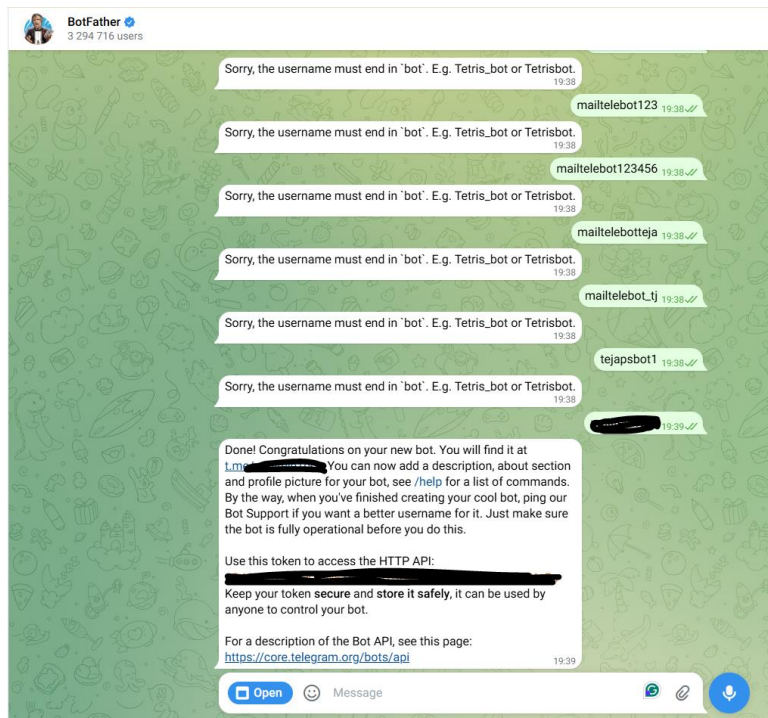
Open bot father

/start

/newbot

<provide username>

And we get our token and bot link.



### Cmd to delete webhook:

- `iwr -Method Post`  
[https://api.telegram.org/bot<YOUR\\_BOT\\_TOKEN>/deleteWebhook](https://api.telegram.org/bot<YOUR_BOT_TOKEN>/deleteWebhook)"

### Cmd to get webhook: iwr

- `iwr https://api.telegram.org/bot<YOUR\_BOT\_TOKEN>/getWebhookInfo`

### Script to setup webhook manually:

```
1. $token = "<YOUR_BOT_TOKEN>"
2. $whUrl = "https://75f1507ee3de.ngrok-free.app/webhook/<YOUR-N8N-WORKFLOW-ID>/webhook"
3. $body = @{
4.     url = $whUrl
5.     allowed_updates = ['"message","callback_query"]'
6. }
7. iwr -Method Post -Body $body -ContentType "application/x-www-form-urlencoded"
https://api.telegram.org/bot$token/setWebhook
```

### To see webhook status:

```
1. $resp = Invoke-RestMethod "https://api.telegram.org/bot$token/getWebhookInfo"
2. $resp | ConvertTo-Json -Depth 6

3. OR

4. (Invoke-WebRequest "https://api.telegram.org/bot$token/getWebhookInfo").Content
```

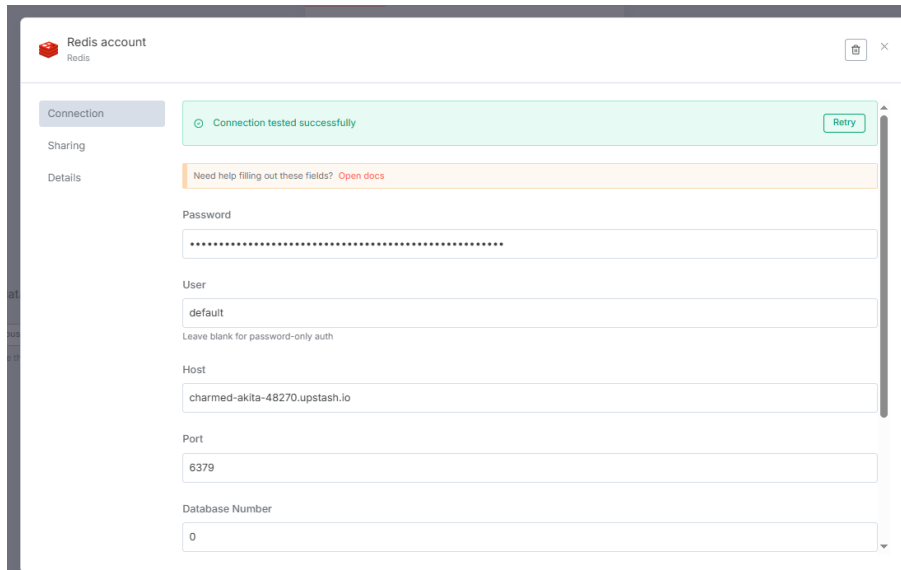
## Step 8: Integrate with Redis:

8.1: Go to <https://console.upstash.com/>

Create redis database

Get tcp connection url

And enter it in our redis node with SSL enabled



Step 9: Tunneling: using ngrok

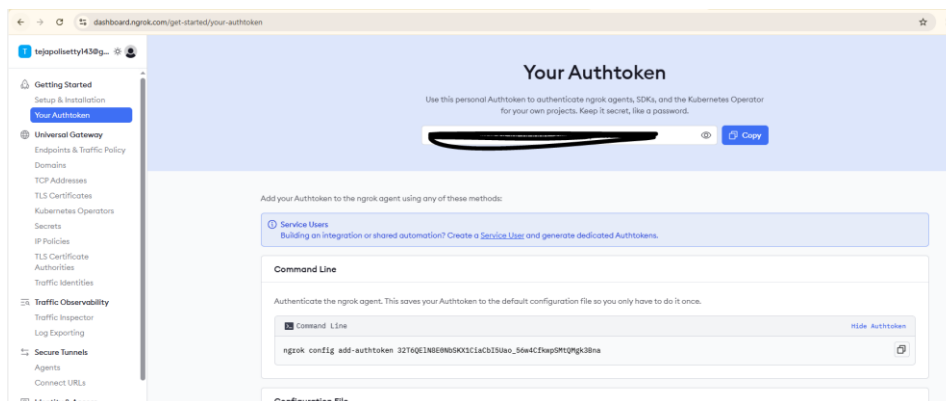
As webhooks only work with https site and not http

So we have to run our localhost on https or host it on some cloud platform

With tunneling, we are running our localhost on https using ngrok

Site: <https://dashboard.ngrok.com/>

Sign in and Get your token:



Cmd for setting auth token:

- `ngrok config add-authtoken <YOUR_AUTHTOKEN>`

Cmd for tunneling:

- `ngrok http 5678 --log=stdout`

But instead we need to stop our docker , and run it with below command to run it on expected port and expected https ngrok url

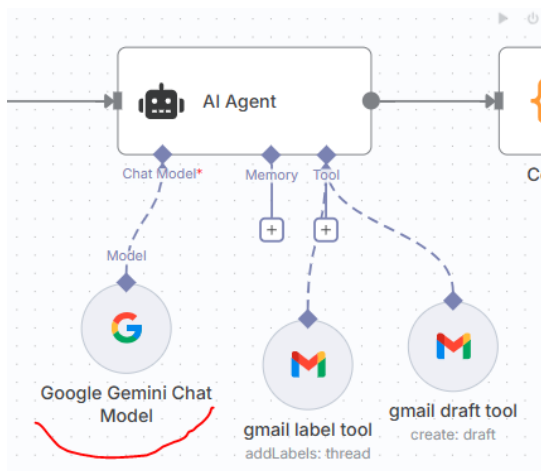
- `docker run -d --name n8n -e N8N_HOST="localhost" -e N8N_PORT=5678 -e N8N_PROTOCOL="http" -e WEBHOOK_URL="https://75f1507ee3de.ngrok-free.app/" -e N8N_EDITOR_BASE_URL="http://localhost:5678/" -p 5678:5678 n8nio/n8n:latest`

## Step 9: Integrate with google gemini:

Site: [aistudio.google.com](https://aistudio.google.com)

Signup And click get API key

In node add agent :



and set account:

The screenshot shows the 'Google Gemini(PaLM) Api account' setup page. The page has a sidebar with 'Connection', 'Sharing', and 'Details' tabs. The 'Connection' tab is active, showing a green status bar with the text 'Connection tested successfully' and a 'Retry' button. Below this, there is a message: 'Need help filling out these fields? [Open docs](#)'. The 'Host' field is filled with 'https://generativelanguage.googleapis.com'. The 'API Key' field is empty, with a placeholder text '.....'. At the bottom, there is a note: 'Enterprise plan users can pull in credentials from external vaults. [More info](#)'.

Step10: Similar to other steps we can get chatgpt openai keys as well.