

How to Set Up an AI Agent for Email Sorting Using n8n and Ollama

(Step-by-Step Guide)

Welcome to this tutorial on how to install and run **n8n** and **Ollama** on your local machine using Docker for the purpose of automated email sorting. This method is easier and avoids dealing with Python, Node.js, or manual backend setup.

Prerequisites

Before you begin, install:

- **Git** – [Download Git](#)
- **Docker Desktop** – <https://www.docker.com/products/docker-desktop/>
 - Download Docker, choose AMD64 version for Windows, and install application

A screenshot of a Windows installation window titled "Installing Docker Desktop 4.43.1 (198352)". The window has a standard Windows title bar with minimize, maximize, and close buttons. The progress bar is partially filled with a blue icon on the left and a green bar in the middle, indicating the installation progress.

Installing Docker Desktop 4.43.1 (198352)

Docker Desktop 4.43.1

Unpacking files...

```
Unpacking file: resources/wsl/docker-wsl-cli.iso
Unpacking file: resources/windows-daemon-options.json
Unpacking file: resources/tile-icon.png
Unpacking file: resources/tile-error.png
Unpacking file: resources/model-runner/bin/com.docker.llama-server.digest
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/docker-desktop.iso.sha256
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
```

- Press the windows key, type cmd, and press enter. Type

`wsl --update`

- Press enter

```
Command Prompt
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KamranE>wsl --update
Checking for updates.
The most recent version of Windows Subsystem for Linux is already installed.

C:\Users\KamranE>
```

Step 2: Setup n8n

- Make sure Docker is running
- Press the windows key
- Type cmd and press enter
- Copy and paste

`docker run -d -p 5678:5678 -v C:\Users\YOURUSER\.n8n:/home/node/.n8n --name n8n n8nio/n8n`

```
C:\Users\KamranE>docker run -it -p 5678:5678 -v C:\Users\KamranE\.n8n:/home/node/.n8n n8nio/n8n
Permissions 0644 for n8n settings file /home/node/.n8n/config are too wide. This is ignored for now, but in the future n8n will attempt to change the permissions automatically. To auto
rce correct permissions now set N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true (recommended), or turn this check off set N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=false.
Initializing n8n process
n8n ready on ::, port 5678
Migrations in progress, please do NOT stop the process.
Starting migration AddInputsOutputsToTestCaseExecution1752669793000
Finished migration AddInputsOutputsToTestCaseExecution1752669793000

There is a deprecation related to your environment variables. Please take the recommended actions to update your configuration:
- N8N_RUNNERS_ENABLED -> Running n8n without task runners is deprecated. Task runners will be turned on by default in a future version. Please set 'N8N_RUNNERS_ENABLED=true' to enable
now and avoid potential issues in the future. Learn more: https://docs.n8n.io/hosting/configuration/task-runners/

[license SDK] attempting license renewal
[license SDK] license successfully renewed
Version: 1.105.4

Editor is now accessible via:
http://localhost:5678

Press "o" to open in Browser.
```

- IMPORTANT: Replace YOURUSER with your windows user
- Go to <http://localhost:5678> in browser

- Set up email username and password
 - Set up free license key (check email)
-

Step 3: Download Ollama

- In command prompt, type
docker run -d --name ollama -p 11434:11434 -v ollama-data:/root/.ollama ollama/ollama
- And press enter

```
C:\Users\KamranE>docker run -d --name ollama -p 11434:11434 -v ollama-data:/root/.ollama ollama/ollama
95e016bbc4244deb94be66e33d654cef7e9aa012ce32accda4e87dcb2a61dc54
```

- In command prompt, type
docker exec -it ollama ollama pull MODELNAME
- Replace MODELNAME with llama3 or any other chosen model

```
C:\Users\KamranE>docker exec -it ollama ollama pull llama3
pulling manifest
pulling 6a0746a1ec1a: 13% | 597 MB/4.7 GB 37 MB/s 1m47s|
```

Step 4: Create gmail credential

- <https://console.cloud.google.com/?pli=1&inv=1&inv=Ab42ew> In your web browser
- Sign into google account
- Select a project > new project
 - Name it whatever you'd like

New Project



You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[Manage Quotas](#)

Project name *

Kam's project



Project ID: kams-project-468316. It cannot be changed later. [Edit](#)



Location *
No organization

[Browse](#)

Parent organization or folder

Create

Cancel

- Click Apis and Services
 - Click Enable API and services
 - Search Gmail API and click enable



Google Cloud



Kam's project



Product details



Gmail API

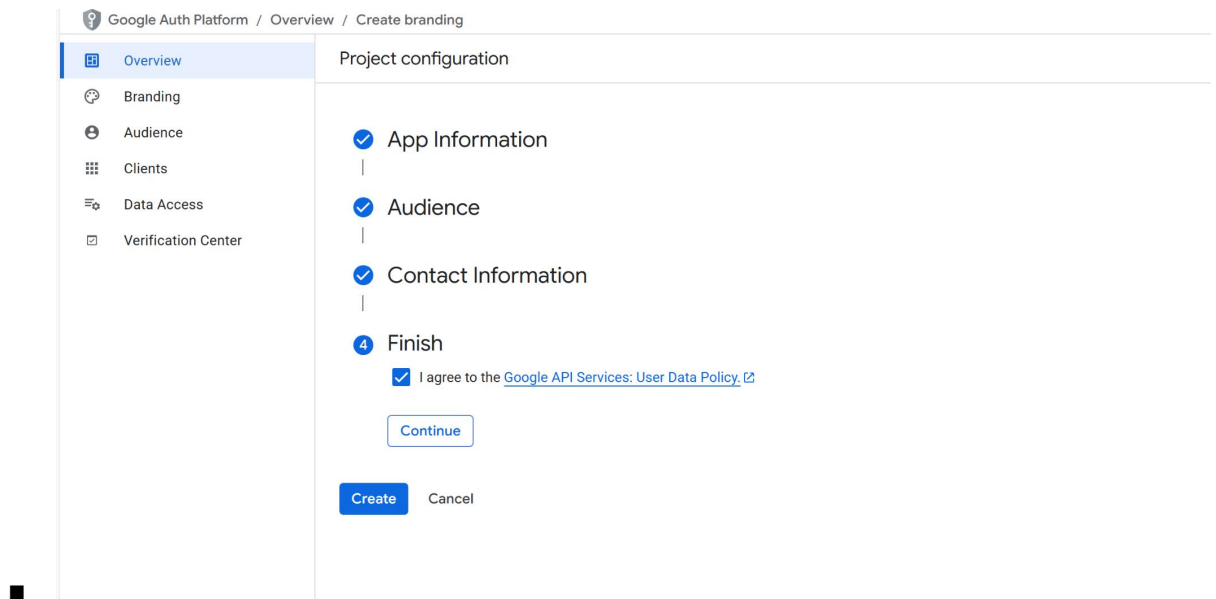
[Google Enterprise API](#)

View and manage Gmail mailbox data.

Enable

[Try this API](#)

- Click Credentials > Add new credentials
- Create Credentials > OAuth client
 - Configure consent screen

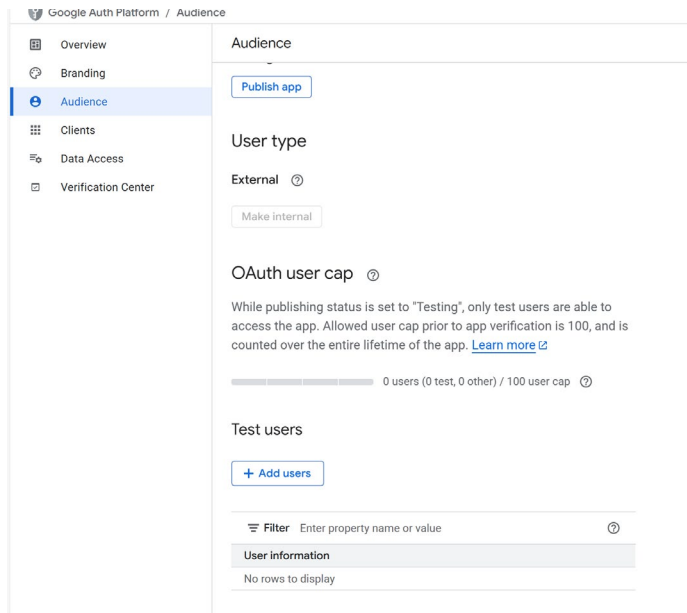


- Click create OAuth client

- Select "Desktop App"

- IMPORTANT: Copy paste Client ID and Client Secret into a text file and save it

- Audience > Add test users > add your email



Step 5: Create Workflow

New label

Please enter a new label name:

Work

☐ Nest label under:

Cancel

Create

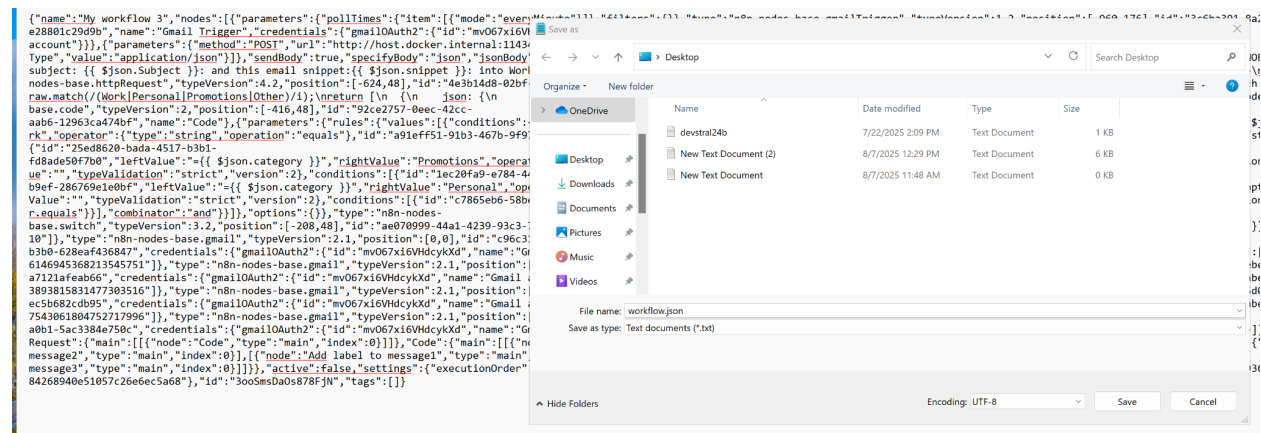
- Go to gmail.com, and by Labels, click plus
 - Add labels: Work, Other, Promo, Personal
- On desktop, right click and create new text file
- Copy and paste the following:

```
{"name":"My workflow
3","nodes":[{"parameters":{"pollTimes":{"item":{"mode":"everyMinute"}},"filters":{},"type":"n8n-nodes-
base.gmailTrigger","typeVersion":1.2,"position":[-960,176],"id":"3c6ba391-8a2e-4ffc-ac71-
e28801c29d9b","name":"Gmail Trigger","credentials":{"gmailOAuth2":{"name":"Gmail
```

```
account"}}, {"parameters": {"method": "POST", "url": "http://host.docker.internal:11434/api/chat", "sendHeaders": true, "headerParameters": {"parameters": [{"name": "Content-Type", "value": "application/json"}]}, "sendBody": true, "specifyBody": "json", "jsonBody": "{\n  \"model\": \"llama3\",\n  \"messages\": [\n    {\n      \"role\": \"user\",\n      \"content\": \"GIVE A ONE WORD RESPONSE:: Classify this email subject: {{ $json.Subject }}: and this email snippet:{{ $json.snippet }}: into Work, Personal, Promotions, Other. Make sure answer is ONLY one of those 4 options.\\n\\n  }\\n  ],\n  \"stream\": false\n}", "options": {}, "type": "n8n-nodes-base.httpRequest", "typeVersion": 4.2, "position": [-624, 48], "id": "4e3b14d8-02bf-419d-8b78-7156b7b3d553", "name": "HTTP Request"}, {"parameters": {"jsCode": "const raw = $json.message.content;\nconst match = raw.match(/(Work|Personal|Promotions|Other)/i);\nreturn [\n  {\n    json: {\n      category: match ? match[1] : 'Other',\n      id: $('Gmail Trigger').first().json.id\n    }\n  }\n];", "type": "n8n-nodes-base.code", "typeVersion": 2, "position": [-416, 48], "id": "92ce2757-0eec-42cc-aab6-12963ca474bf", "name": "Code"}, {"parameters": {"rules": {"values": [{"conditions": {"options": {"caseSensitive": true, "leftValue": "", "typeValidation": "strict", "version": 2}, "conditions": [{"leftValue": "{{ $json.category }}", "rightValue": "Work", "operator": {"type": "string", "operation": "equals"}}, {"combinator": "and"}], "conditions": {"options": {"caseSensitive": true, "leftValue": "", "typeValidation": "strict", "version": 2}, "conditions": [{"leftValue": "{{ $json.category }}", "rightValue": "Promotions", "operator": {"type": "string", "operation": "equals"}}, {"combinator": "and"}], "conditions": {"options": {"caseSensitive": true, "leftValue": "", "typeValidation": "strict", "version": 2}, "conditions": [{"leftValue": "{{ $json.category }}", "rightValue": "Personal", "operator": {"type": "string", "operation": "equals"}}, {"combinator": "and"}], "conditions": {"options": {"caseSensitive": true, "leftValue": "", "typeValidation": "strict", "version": 2}, "conditions": [{"leftValue": "{{ $json.category }}", "rightValue": "Other", "operator": {"type": "string", "operation": "equals"}}, {"combinator": "and"}]}], "options": {"type": "n8n-nodes-base.switch", "typeVersion": 3.2, "position": [-208, 48], "id": "ae070999-44a1-4239-93c3-71f157e9abb4", "name": "Switch"}, {"parameters": {"operation": "addLabels", "messageId": "{{ $('Gmail Trigger').item.json.id }}", "labelNames": ["Work"], "type": "n8n-nodes-base.gmail", "typeVersion": 2.1, "position": [0, 0], "id": "c96c3187-ee9c-480c-a6ec-11d33c8e5643", "name": "Add label to message", "webhookId": "0bc2eaa5-8316-4f0a-b3b0-628eaf436847", "credentials": {"gmailOAuth2": {"name": "Gmail account"}}, {"parameters": {"operation": "addLabels", "messageId": "{{ $('Gmail Trigger').item.json.id }}", "labelNames": ["Promotions"], "type": "n8n-nodes-base.gmail", "typeVersion": 2.1, "position": [144, 112], "id": "9e2f4492-c01d-449d-90cd-8123f385be71", "name": "Add label to message1", "webhookId": "df00eb5c-37bd-4c89-8ebe-a7121afeab66", "credentials": {"gmailOAuth2": {"name": "Gmail account"}}, {"parameters": {"operation": "addLabels", "messageId": "{{ $('Gmail Trigger').item.json.id }}", "labelNames": ["Personal"], "type": "n8n-nodes-base.gmail", "typeVersion": 2.1, "position": [240, 240], "id": "db2785b8-2475-4460-91ef-10803021145b", "name": "Add label to message2", "webhookId": "183aab7c-e804-4459-a5d0-ec5b682cdb95", "credentials": {"gmailOAuth2": {"name": "Gmail account"}}, {"parameters": {"operation": "addLabels", "messageId": "{{ $('Gmail Trigger').item.json.id }}", "labelNames": ["Other"], "type": "n8n-nodes-base.gmail", "typeVersion": 2.1, "position": [0, 256], "id": "749a2585-88db-454d-8605-a24d58572a61", "name": "Add label to message3", "webhookId": "6119a8ac-4aee-4f15-a0b1-
```

```
5ac3384e750c","credentials":{"gmailOAuth2":{"name":"Gmail
account"}}},"pinData":{"connections":{"Gmail Trigger":{"main":[{"node":"HTTP
Request","type":"main","index":0}]},"HTTP
Request":{"main":[{"node":"Code","type":"main","index":0}]},"Code":{"main":[{"node":"Switch","type":"m
ain","index":0}]},"Switch":{"main":[{"node":"Add label to
message","type":"main","index":0},{"node":"Add label to
message2","type":"main","index":0},{"node":"Add label to
message1","type":"main","index":0},{"node":"Add label to
message3","type":"main","index":0}]}]},"active":false,"settings":{"executionOrder":"v1"},"versionId":"22c56
a19-2848-4d57-92f3-
2bbce8a3c48e","meta":{"templateCredsSetupCompleted":true,"instanceId":"1ed93660c3afa5c476205a8
7cbf702cf51ac72184268940e51057c26e6ec5a68"},"id":"3ooSmsDaOs878FjN","tags":[]}
```

- File > Save As > workflow.json



- In n8n (<http://localhost:5678> in browser), click create new workflow
- Click the three dots at the top right, and import from file
- Select workflow.json from desktop
- Go to gmail node:
 - Add new credential

- Paste previously saved id and secret into respective fields
- Sign in with same google account previously added as test user

Gmail Trigger Fetch Test Event

Parameters Settings Docs

Credential to connect with
 Gmail account

Poll Times

Mode
 Every Minute

Add Poll Time

Event

- Double click every node once and click execute workflow
- Then, double click every add label node and make sure your credential is selected

Add label to message

Parameters Settings

Credential to connect with
 Gmail account

Resource
 Message

Operation

- Ensure each label node has its respective label added under “Label Names or IDs”
- KEY:
 - Message - Work
 - Message2 - Promo
 - Message1 - Personal
 - Message 3 - Other

- Click Execute Workflow, and enjoy automatically sorted emails (set to run every 5 minutes while open)

Troubleshooting

n8n does not open in browser (<http://localhost:5678>):

- Make sure the container is running by typing in command prompt:
docker ps
- If you don't see n8n listed, run the setup command:
docker run -d -p 5678:5678 -v C:\Users\YOURUSER\.n8n:/home/node/.n8n --name n8n n8nio/n8n
- Replace YOURUSER with your windows user

Ollama API call fails in HTTP Request node:

- Make sure ollama is running by typing om command prompt
docker ps
- If it's not running, start it using:
docker start ollama
- Make sure you've pulled the model by typing:
docker exec -it ollama ollama pull llama3

Gmail OAuth credentials not working:

- Make sure you selected "Desktop App" when creating the OAuth client
- Make sure you copied the Client ID and Client Secret correctly
- In Google Cloud Console, go to OAuth consent screen > Test Users, and make sure your Gmail address is added

Conclusion

Using Docker with n8n and Ollama is the easiest way to run automated Gmail sorting locally on your machine.

Written by **Kamran Eisenberg**
For more guides, visit [[SolutionsInfra.com](https://solutionsinfra.com)]