**Step-by-Step Guide to Create and Deploy a Webhook Server using Flask**

### 1. Set Up Flask Webhook Server Locally

First, you'll need to install Python and Flask, then create a simple Flask app to handle Instagram webhook events.

**Install Flask**

Make sure you have Python installed, and then install Flask and any other necessary libraries:

###### pip install Flask gunicorn

### Create the Flask Webhook Server

Create a new Python file, for example, app.py:

from flask import Flask, request, jsonify

app = Flask(\_\_name\_\_)

# Verification Token

VERIFY\_TOKEN = "my\_secure\_token\_123"

# Handle webhook verification (GET request)

@app.route('/webhook', methods=['GET'])

def verify\_webhook():

    mode = request.args.get('hub.mode')

    token = request.args.get('hub.verify\_token')

    challenge = request.args.get('hub.challenge')

    if mode and token:

        if mode == 'subscribe' and token == VERIFY\_TOKEN:

            print("WEBHOOK VERIFIED")

            return challenge, 200

        else:

            return "Forbidden", 403

# Handle webhook events (POST request)

@app.route('/webhook', methods=['POST'])

def handle\_webhook():

    data = request.get\_json()

    if data['object'] == 'instagram':

        print('Received Instagram event:', data)

        # Process the event here

    return "EVENT RECEIVED", 200

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True, port=5000)

In this example:

* **GET /webhook**: Verifies the webhook with the token you provide in your Facebook Developer Dashboard.
* **POST /webhook**: Handles incoming webhook events (messages, comments, etc.).

### 2. Test the Webhook Locally

To test your Flask app locally, run the Flask server:

###### python app.py

The server will run at http://localhost:5000/webhook. However, Instagram requires your webhook to be publicly accessible over HTTPS, so you’ll need to expose your local server.

You can use **ngrok** to expose your local server:

**What ngrok Does:**

* **ngrok** creates a secure tunnel from your machine (localhost) to a public URL that anyone (including Instagram) can access.
* This public URL is randomly generated, meaning each time you run ngrok, you get a new URL.

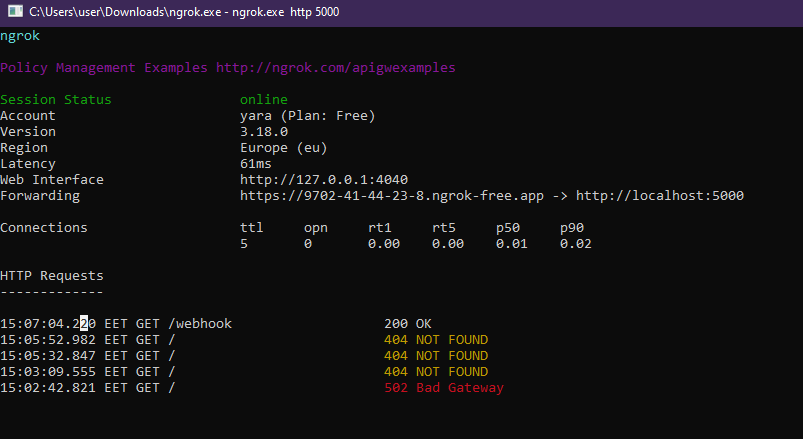
**Steps to Use ngrok for Webhooks:**

**1. Install ngrok:**

If you don’t have ngrok installed yet, you can download it from ngrok's official website.

Run:

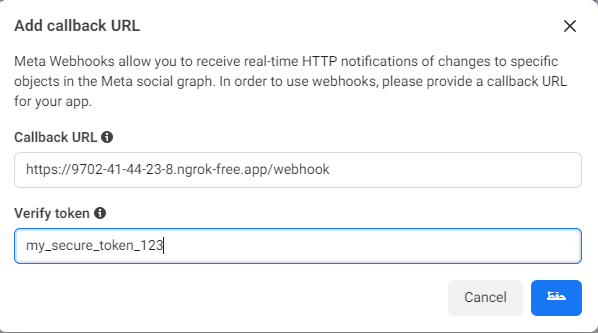
ngrok http 5000

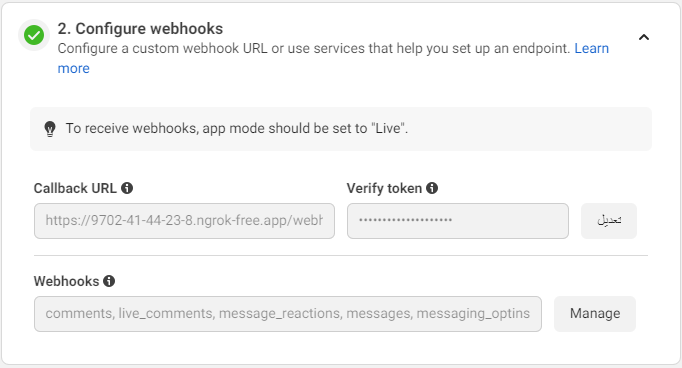


This will give you a public URL (e.g., https://<random-string>.ngrok.io/webhook) that you can use temporarily as your callback URL in my case I used https://9702-41-44-23-8.ngrok-free.app/webhook

**3. Set the Callback URL in Facebook Developer Dashboard**

* Go to the [Facebook Developer Dashboard](https://developers.facebook.com/).
* Under your app, navigate to **Webhooks**.
* Select **Instagram** from the available Webhooks options.
* Add your **callback URL** (e.g., https://<random-string>.ngrok.io/webhook).
* Add your **verification token** (make sure it matches VERIFY\_TOKEN in your code).





Facebook will verify the webhook by sending a GET request to your server with a challenge. Your Flask server will respond, and Facebook will confirm if the webhook is successfully verified.

**Server deployment**

**Free option:**

**Vercel**

I used vercel to deploy my server and now it can be accessed through a permanent link, all you have to do is upload your code on github and link the project to vercel through guided steps.

**My server link**: <https://cbs-beta.vercel.app/>

**Github link:** https://github.com/dataheadway/CBS

**Tutorial:** https://youtu.be/miQmOlPF\_Gs

**A screenshot of a computer

Description automatically generated**

**A screenshot of a webbook

Description automatically generated**

**Paid option:**

- **Heroku**:

* **Free tier**: Basic apps with limitations.
* **Hobby dynos**: $7/month for small apps.
* **Production dynos**: Start at $25/month.
* **Performance dynos**: For high-demand apps, up to $500+/month depending on resources.
* Additional services like databases and memory also scale with costs​

A screenshot of a computer

Description automatically generated

A screenshot of a web page

Description automatically generated

- **Amazon Web Services (AWS)**:

* **Pay-as-you-go** model: Costs vary greatly based on usage (compute, storage, etc.).
* **EC2 instances**: Small instances can start as low as $3.50/month, but larger and high-performance instances can go beyond hundreds per month.
* **Free tier**: 12 months of free limited services (750 hours/month for EC2 micro-instances, etc.)​

- **Google Cloud Platform (GCP)**:

* **Free tier**: Includes $300 in free credits and limited free services.
* **Pricing**: GCP pricing is similar to AWS, but often offers more flexibility for custom virtual machines. Smaller VMs can cost around $4-5/month.
* **Compute Engine**: Offers preconfigured and custom machine types with prices scaling depending on resource needs​.

- **DigitalOcean**:

* **Droplets (VMs)**: Start at $4/month for the most basic configuration.
* **App Platform**: Starts at $5/month for a basic plan and scales based on usage.
* **Other Services**: Kubernetes ($12/month), block storage ($10/month), and load balancers ($12/month)​

Screens screenshot of a blue and white website

Description automatically generated

A screenshot of a website

Description automatically generatedA screenshot of a screenshot of a website

Description automatically generated