Hey! Here’s a guide to get your own version of the Telegram tracker bot up and running. Just follow these steps.

### **## Step 1: Fork the Code on GitHub**

**What you're doing:** Creating a personal copy of the bot's source code in your own GitHub account. This allows you to connect it to Render.

**Where to do it:** In your web browser, on the GitHub website.

1. Log in to your own GitHub account.
2. Go to the project's GitHub page:<https://github.com/markhendriksen1990/Nodify.git>
3. In the top-right corner of the page, click the **Fork** button. This will create an exact copy of the project under your own GitHub account.

### **## Step 2: Create Your Telegram Bot**

**What you're doing:** Registering your bot with Telegram to get an API token, which is like a password for your code to control the bot.

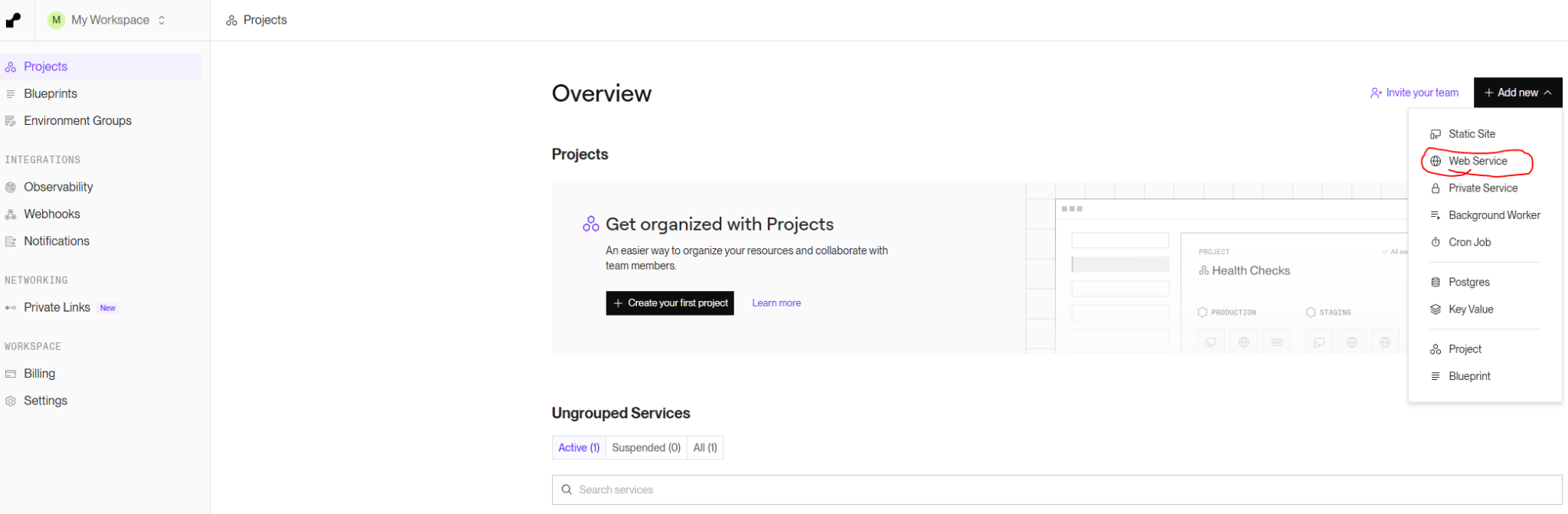
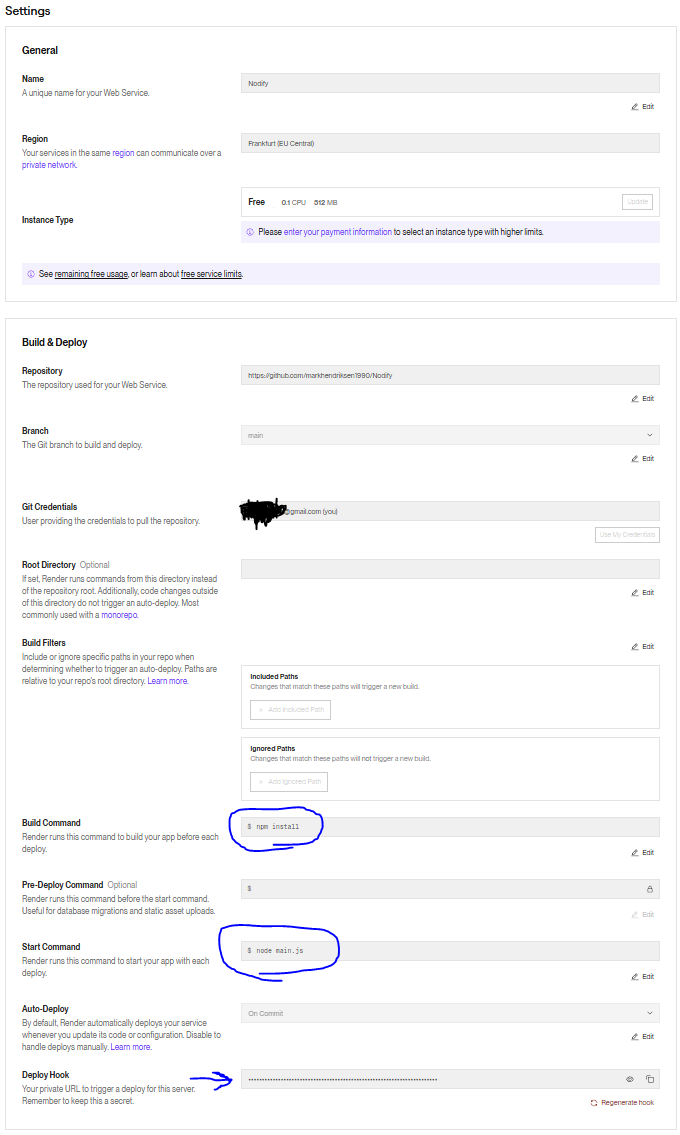
**Where to do it:** In the Telegram app, by talking to a bot called BotFather.

1. In Telegram, search for **BotFather** (it has a blue checkmark) and start a chat with it.
2. Send the command /newbot.
3. BotFather will ask for a name for your bot (e.g., "My DeFi Tracker") and then a username, which must be unique and end in bot (e.g., MyDeFiTrackerBot).
4. Once you're done, it will give you a message containing your **bot token**. It looks something like 7572395356:AAHVp1XBZzqi7Eoz3TzUDjDKui0beQAQyPE.
5. **Copy this token and save it somewhere safe.** You'll need it in a later step. This is your  
    TELEGRAM\_BOT\_TOKEN.
6. When a new chat c.q. bot is created you have to find it’s chat ID  
   To get it, send /start to the @userinfobot on Telegram and copy the ID it gives you. **Copy this chat ID and save it somewhere safe.** It looks something like this: 540263811  
   You'll need it in a later step. This is your CHAT\_ID.

### **## Step 3: Deploy the Bot on Render**

**What you're doing:** Setting up a free web server on Render that will run your bot's code 24/7.

**Where to do it:** In your web browser, on the [Render dashboard](https://dashboard.render.com/).

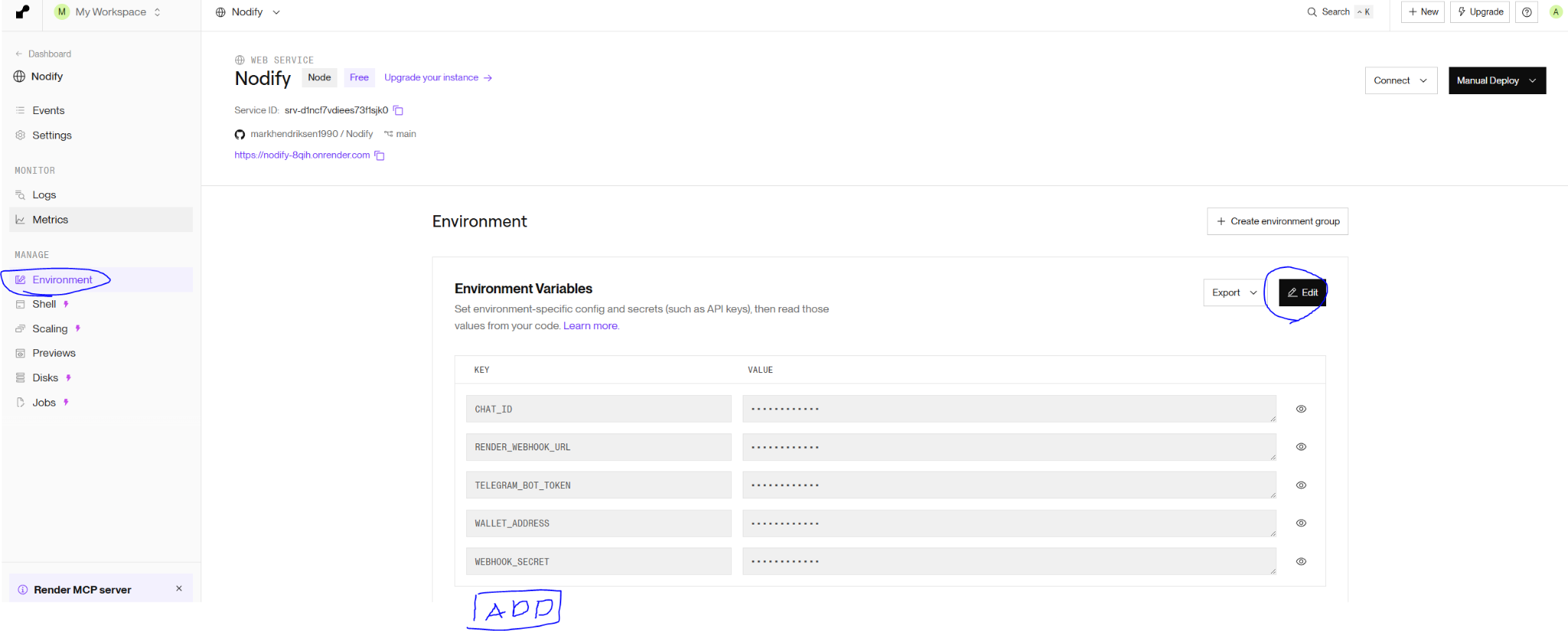
1. Create a Render account. It's easiest to sign up using your GitHub account.
2. On the dashboard, click **New +** and select **Web Service**.  
   ****
3. Connect the GitHub repository that you forked in Step 1.
4. Render will ask you to configure the service. Use these settings:
   * **Name**: Give your service a unique name (e.g., nodify-yourname).
   * **Runtime**: Render should automatically detect Node.
   * **Build Command**: npm install.
   * **Start Command**: node main.js.
   * **Instance Type**: Select the **Free** tier.  
     
5. Before creating the service, scroll down to the **Environment Variables** section. This is where you'll put your secrets.

### **## Step 4: Add Your Environment Variables**

**What you're doing:** Giving your Render service the secret keys and configuration it needs to run.

**Where to do it:** In the "Environment" section during the Render setup process.

1. Click **Add Environment Variable** and add the following key-value pairs one by one:
   * **Key**: TELEGRAM\_BOT\_TOKEN
     + **Value**: The token you got from BotFather in Step 2.
   * **Key**: CHAT\_ID
     + **Value**: See Step 2 - create telegram bot.
   * **Key**: RENDER\_WEBHOOK\_URL
     + **Value**: Your service's public URL. Render gives you this once it's created. It will look like  
        https://your-service-name.onrender.com.
   * **Key**: WALLET\_ADDRESS
     + **Value**: The public Ethereum wallet address you want the bot to monitor (e.g., 0xAb1234a7D312341b8bE11C439e05C5A3259aeC9A).  
       **You can add multiple addresses using “,”.**  
       For instance   
       0xAb…1111aeC9B,0xAb…eC9B,0xAb…eC9C
   * **Key**: WEBHOOK\_SECRET
     + **Value**: A random, secure password you create yourself. This is used to make sure the requests are coming from Telegram. Example:  
       MySuperSecretPassword123!.



1. Once all variables are added, click **Create Web Service**. Render will now build and deploy your bot. This might take a few minutes.

### **## Step 5: Connect Telegram to Your Bot on Render**

**What you're doing:** Telling Telegram to send all messages and commands for your bot to your new app running on Render. This is called setting the "webhook."

**Where to do it:** In your web browser's address bar.

Construct the following URL by replacing the parts in  
 <...> with your actual values from the previous steps.  
https://api.telegram.org/bot<YOUR\_TELEGRAM\_BOT\_TOKEN>/setWebhook?url=<YOUR\_RENDER\_WEBHOOK\_URL>/bot<YOUR\_TELEGRAM\_BOT\_TOKEN>/webhook&secret\_token=<YOUR\_WEBHOOK\_SECRET>

1. **Example:**

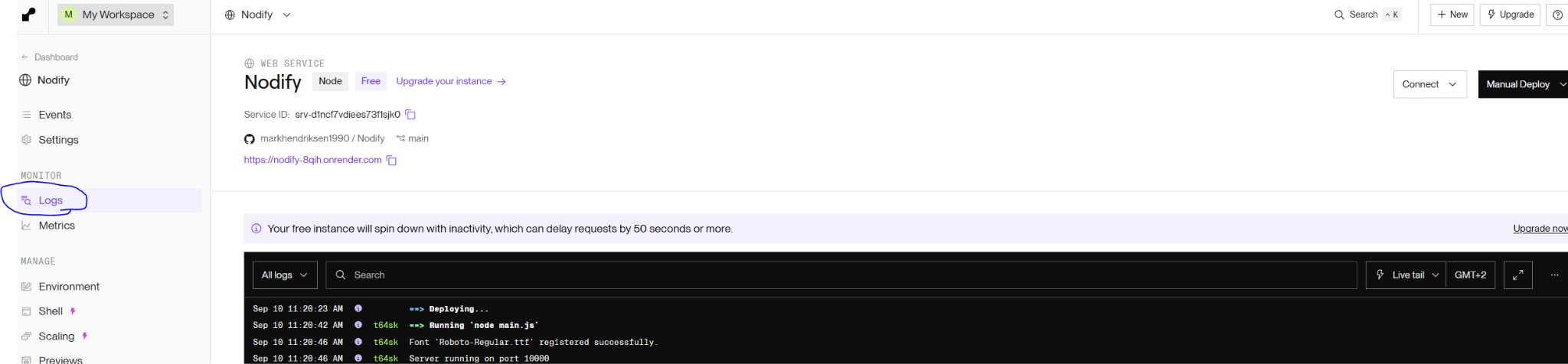
https://api.telegram.org/bot1234123456:ABHVpiXBZsqi7Eoz5TzUDjDKwi3beQAQyPE/setWebhook?url=https://nodify-1q1v.onrender.com/bot1234123456:ABHVpiXBZsqi7Eoz5TzUDjDKwi3beQAQyPE/webhook&secret\_token=MySuperSecretPassword123!

1. Copy your fully constructed URL, paste it into your browser's address bar, and press **Enter**.
2. If it's successful, you will see a simple message in your browser like: {"ok":true,"result":true,"description":"Webhook was set"}
3. In case needed:  
   https://api.telegram.org/bot1234123456:ABHVpiXBZsqi7Eoz5TzUDjDKwi3beQAQyPE/getWebhookInfo  
   This gives you it’s status
4. https://api.telegram.org/bot1234123456:ABHVpiXBZsqi7Eoz5TzUDjDKwi3beQAQyPE/deleteWebhook  
   This deletes the telegram webhook link.  
   **Sometimes you have to delete it and connect it after running faulty code to often.**

### **## Final Check**

You're all set! 🚀 Go to your bot in Telegram and send the

/positions or /snapshot command. It might take a moment to respond the first time as it gathers all the data.

If it doesn't work, the first place to check is the **Logs** tab for your service in the Render dashboard. It will usually tell you what went wrong.  


Good luck!

—-------