Delivering Seamless Customer Engagement with WhatsApp Integration

Why WhatsApp for Service Desk Operations?

- **Instant Communication**: WhatsApp allows real-time interaction with customers, reducing response times and improving customer satisfaction.
- **Wide Reach**: With over 2 billion users globally, WhatsApp enables businesses to engage with customers on a platform they already trust and use daily.
- **Automation & Efficiency**: Integrating WhatsApp with backend systems can automate routine tasks, such as ticket creation and status updates, freeing up agents to focus on more complex issues.
- **Rich Media Support**: Share documents, images, and links directly within the chat, providing a more comprehensive support experience.

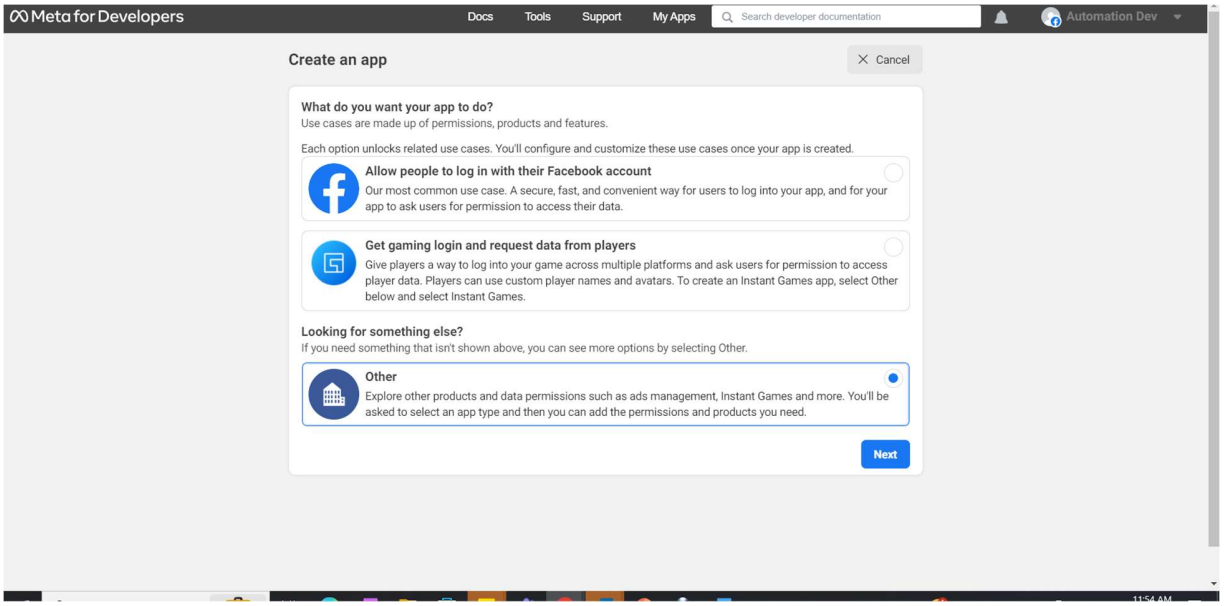
Pre-requisites

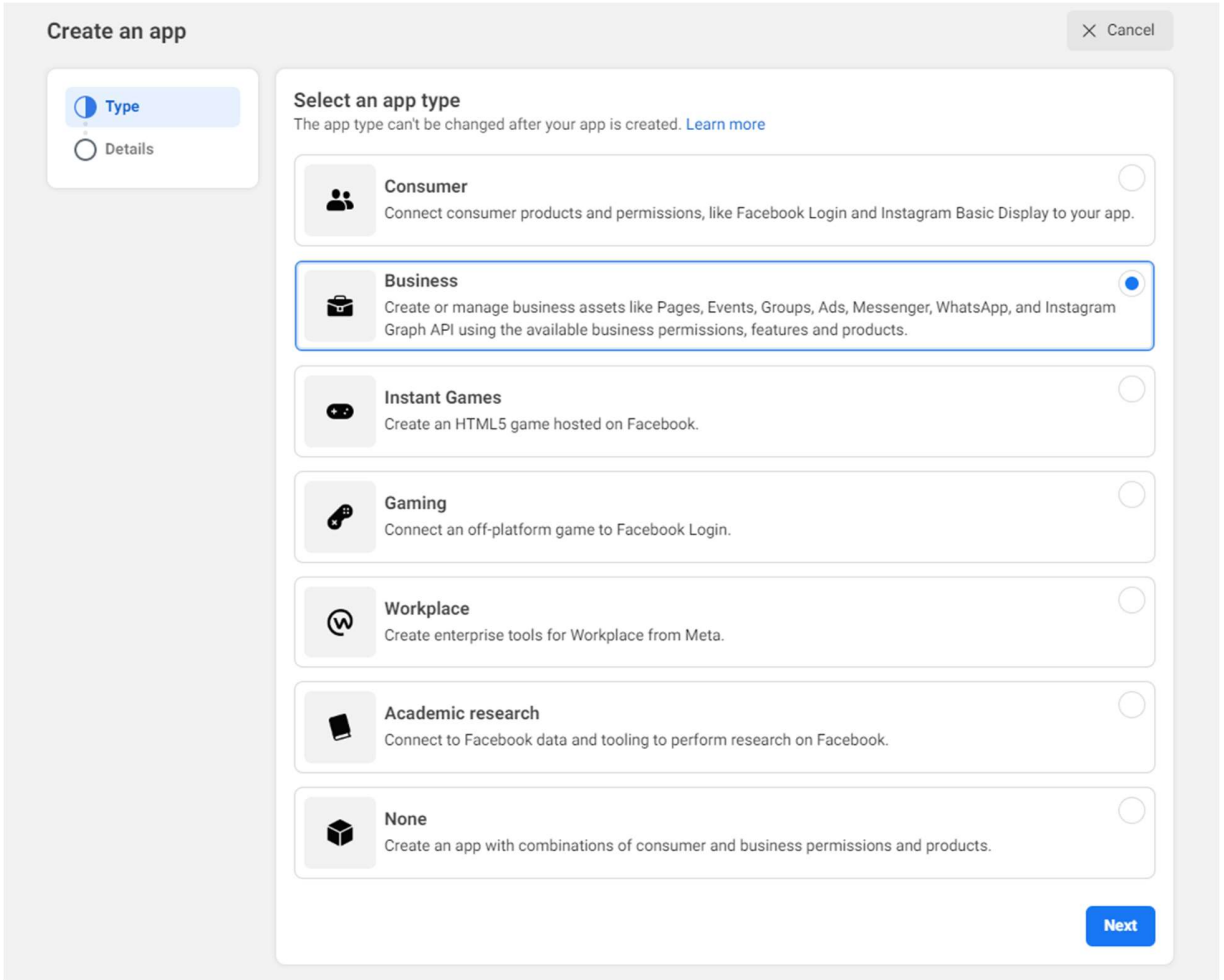
- WhatsApp registered business account phone number
- Meta Business account
- Web server link with SSL enabled – to host the chatbot application
- Backend - Node js

- **WhatsApp registered business account phone number**

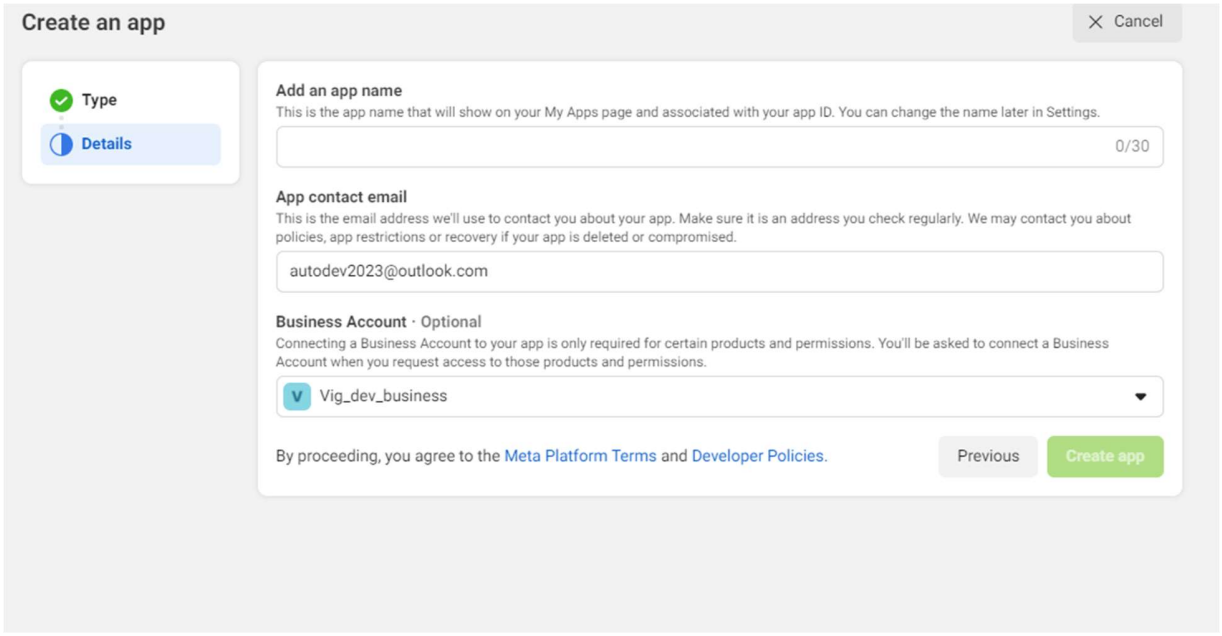
  Create an account in https://developers.facebook.com/

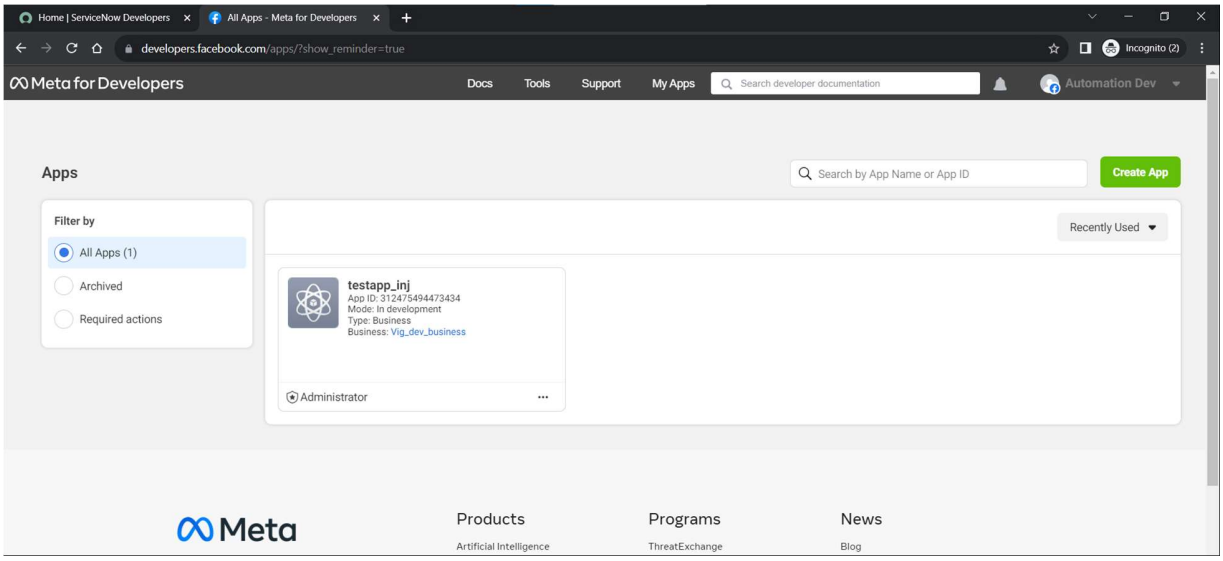
  Create an app

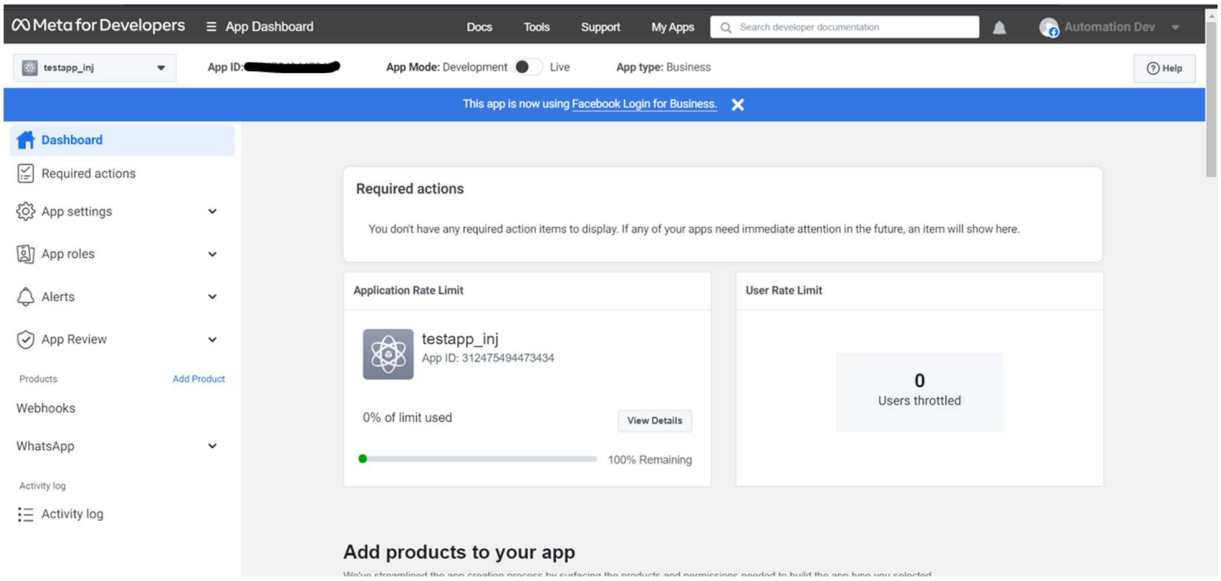

Meta for Developers
Docs
Tools
Support
My Apps
Search developer documentation
Automation Dev
Create an app
Cancel
What do you want your app to do?
Use cases are made up of permissions, products and features.
Each option unlocks related use cases. You'll configure and customize these use cases once your app is created.
Allow people to log in with their Facebook account
Our most common use case. A secure, fast, and convenient way for users to log into your app, and for your app to ask users for permission to access their data.
Get gaming login and request data from players
Give players a way to log into your game across multiple platforms and ask users for permission to access player data. Players can use custom player names and avatars. To create an Instant Games app, select Other below and select Instant Games.
Looking for something else?
If you need something that isn't shown above, you can see more options by selecting Other.
Other
Explore other products and data permissions such as ads management, Instant Games and more. You'll be asked to select an app type and then you can add the permissions and products you need.
Next


Create an app
Cancel
Type
Details
Select an app type
The app type can't be changed after your app is created. Learn more
Consumer
Connect consumer products and permissions, like Facebook Login and Instagram Basic Display to your app.
Business
Create or manage business assets like Pages, Events, Groups, Ads, Messenger, WhatsApp, and Instagram Graph API using the available business permissions, features and products.
Instant Games
Create an HTML5 game hosted on Facebook.
Gaming
Connect an off-platform game to Facebook Login.
Workplace
Create enterprise tools for Workplace from Meta.
Academic research
Connect to Facebook data and tooling to perform research on Facebook.
None
Create an app with combinations of consumer and business permissions and products.
Next

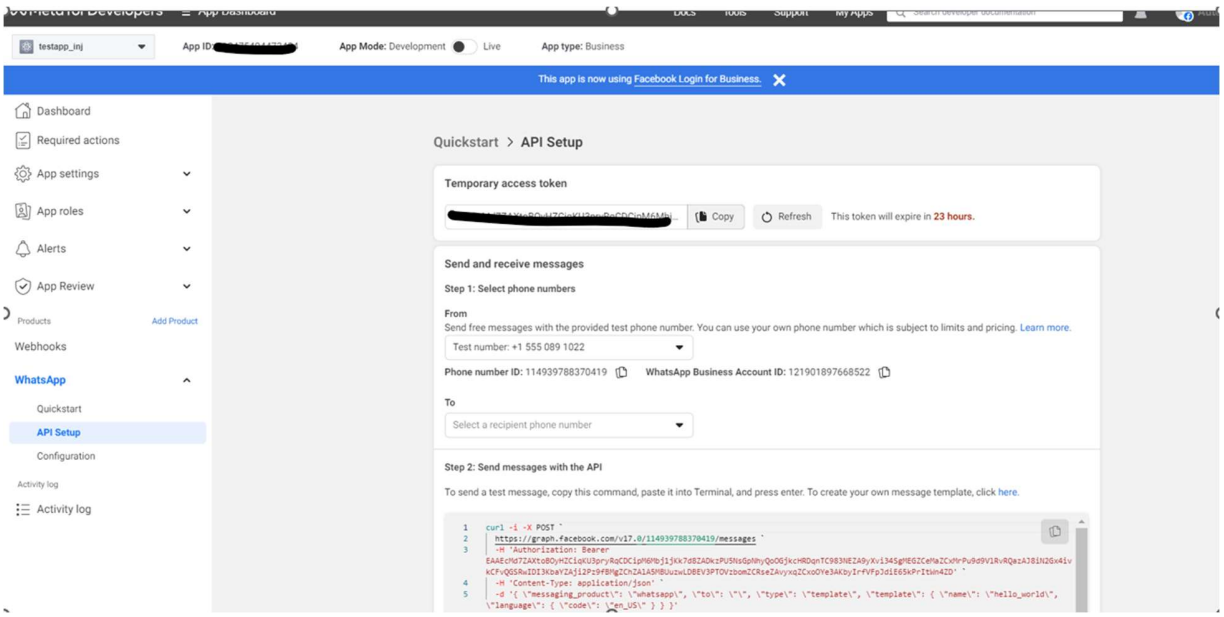
Select / create a new business account

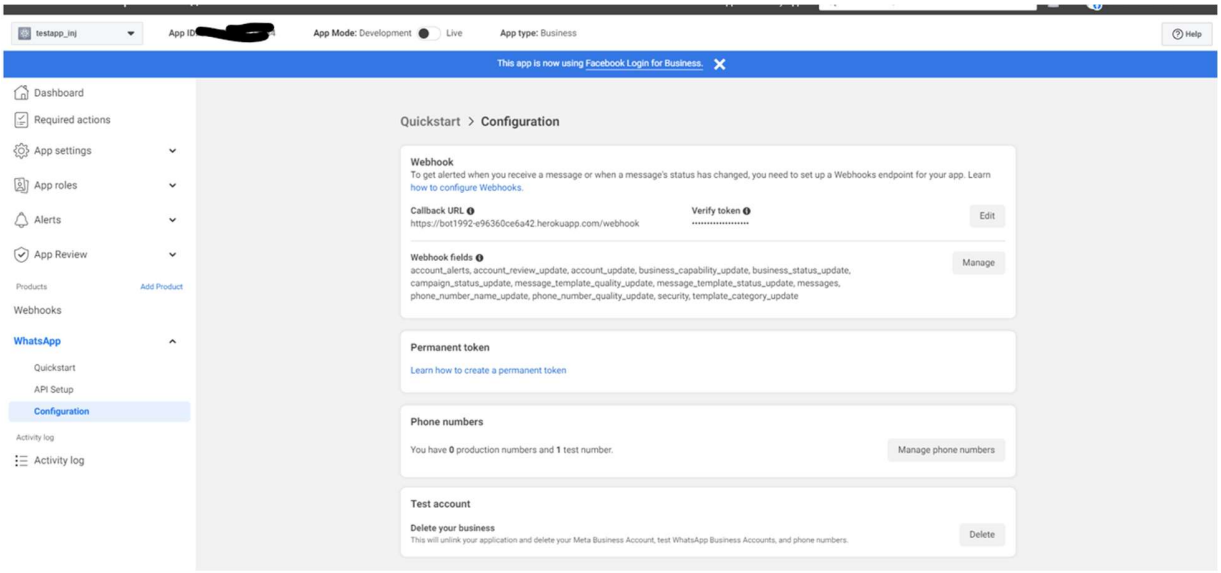


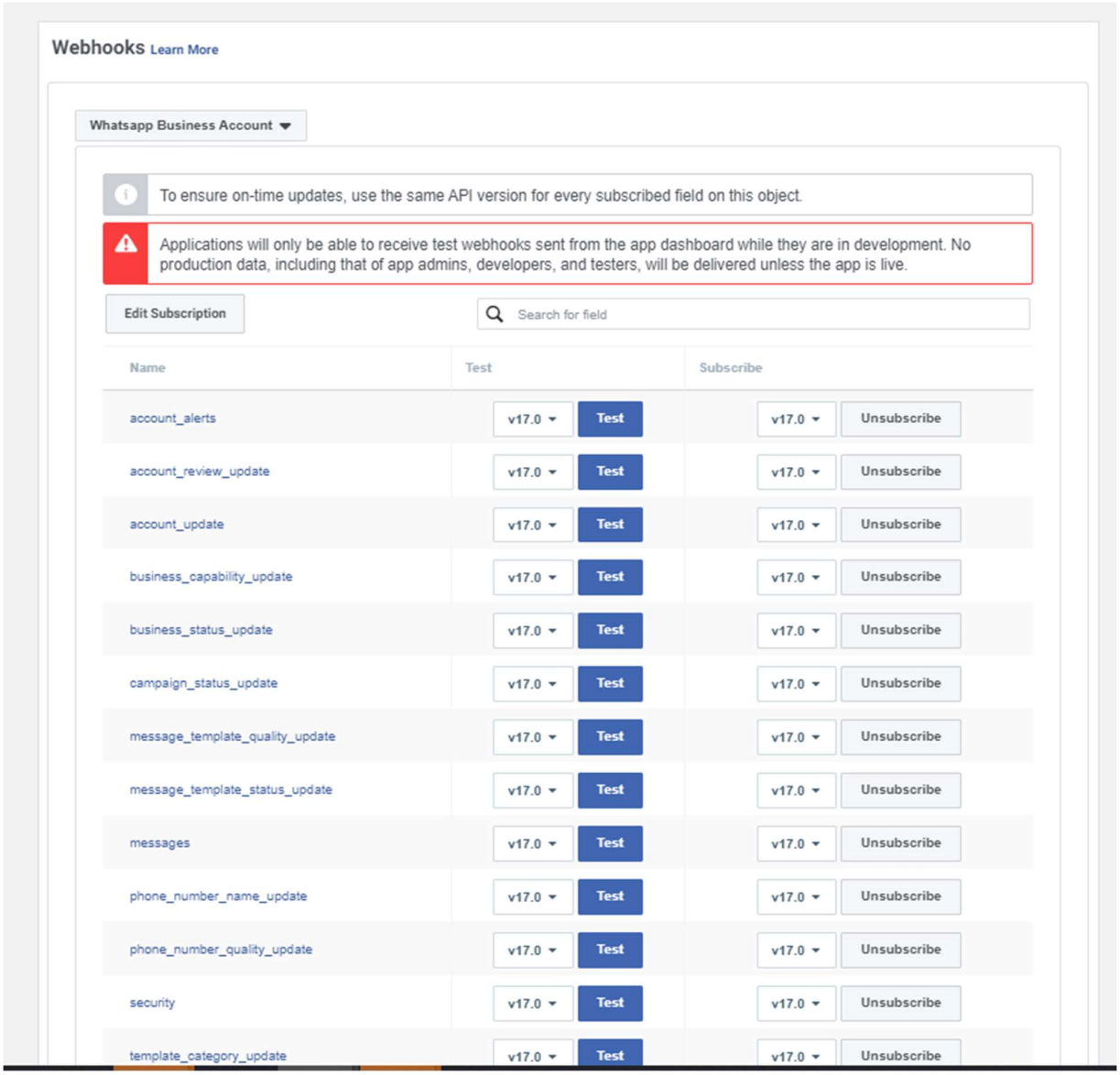




Add Phone numbers

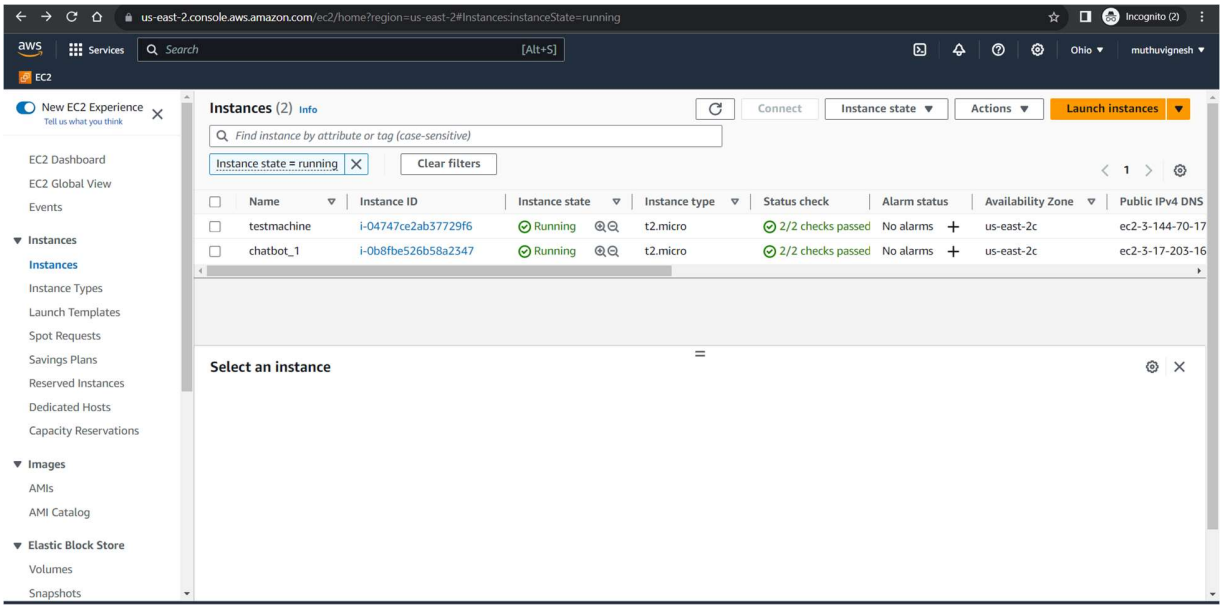


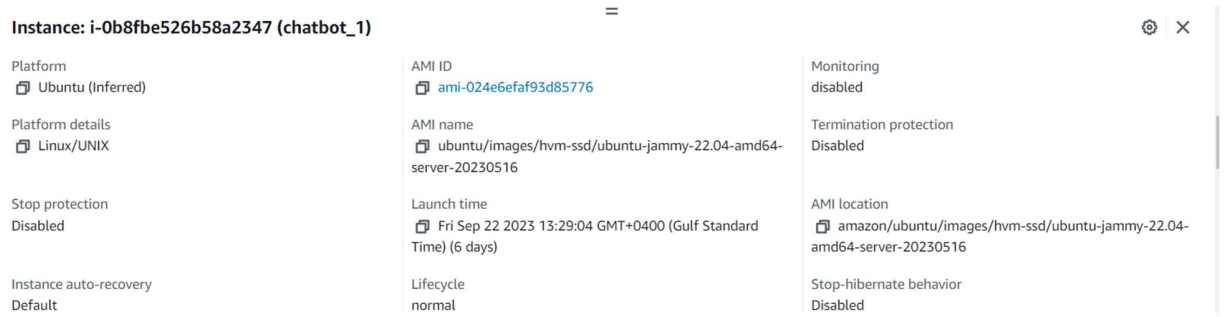

testapp_inj
App ID
App Mode: Development
Live
App type: Business
Help
This app is now using Facebook Login for Business.
Dashboard
Required actions
App settings
App roles
Alerts
App Review
Products
Add Product
Webhooks
WhatsApp
Quickstart
API Setup
Configuration
Activity log
Activity log
Quickstart > Configuration
Webhook
To get alerted when you receive a message or when a message's status has changed, you need to set up a Webhooks endpoint for your app. Learn how to configure Webhooks.
Callback URL
https://bot1992-e96360ce6a42.herokuapp.com/webhook
Verify token
Edit
Webhook fields
account_alerts, account_review_update, account_update, business_capability_update, business_status_update, campaign_status_update, message_template_quality_update, message_template_status_update, messages, phone_number_name_update, phone_number_quality_update, security, template_category_update
Manage
Permanent token
Learn how to create a permanent token
Phone numbers
You have 0 production numbers and 1 test number.
Manage phone numbers
Test account
Delete your business
This will unlink your application and delete your Meta Business Account, test WhatsApp Business Accounts, and phone numbers.
Delete

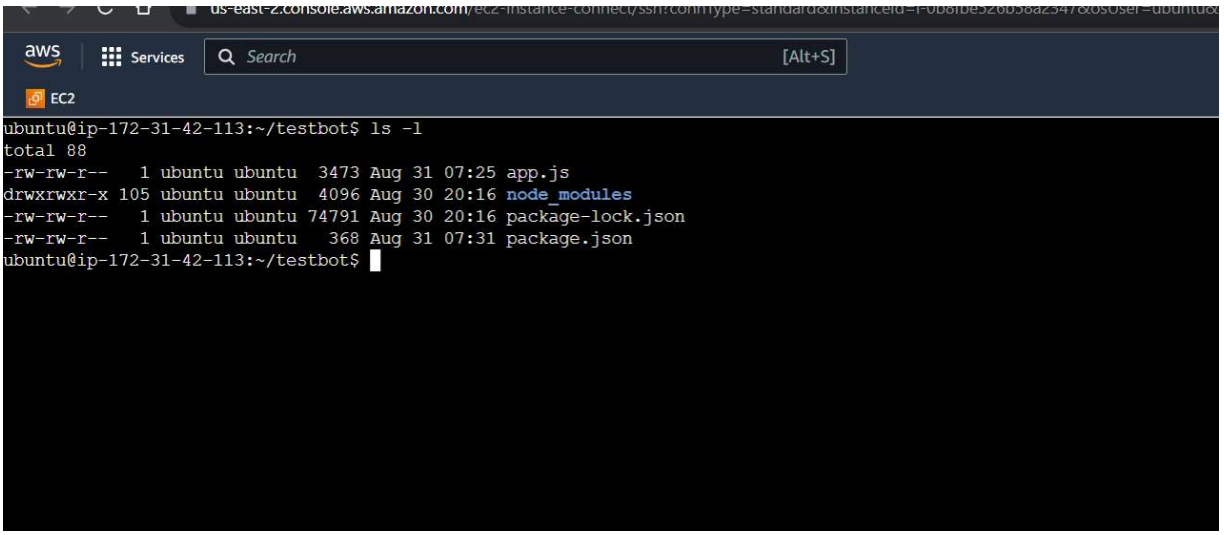


2. Build the app

Login to AWS and create an ubuntu machine







Create a new node project

```
ubuntu@ip-<ipaddress>:~/testbot$ cat package.json
{
  "name": "testbot",
  "version": "1.0.0",
  "description": "sample chatbot with whatsapp",
  "main": "app.js",
  "scripts": {
    "start": "node app.js"
  },
  "author": "Muthu dharmaraj",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.27.2",
    "body-parser": "^1.20.0",
    "dotenv": "^16.3.1",
    "express": "^4.18.1",
    "request": "^2.88.2"
  }
}

ubuntu@ip-<ipaddress>:~/testbot$ cat app.js
/*
 * Starter Project for WhatsApp Echo Bot Tutorial
 *
 * Remix this as the starting point for following the WhatsApp Echo Bot tutorial
 *
 */

"use strict";
require('dotenv').config();
// Access token for your app
// (copy token from DevX getting started page
// and save it as environment variable into the .env file)
const token = process.env.WHATSAPP_TOKEN;
//console.log("test")
//console.log(process.env.WHATSAPP_TOKEN)
//console.log(token)
// Imports dependencies and set up http server
const request = require("request"),
  express = require("express"),
  body_parser = require("body-parser"),
  axios = require("axios").default,
  app = express().use(body_parser.json()); // creates express http server
```

```
// Sets server port and logs message on success
app.listen(process.env.PORT || 1337, () => console.log("webhook is listening"));

// Accepts POST requests at /webhook endpoint
app.post("/webhook", (req, res) => {
 // Parse the request body from the POST
 let body = req.body;

 // Check the Incoming webhook message
 console.log(JSON.stringify(req.body, null, 2));

 // info on WhatsApp text message payload:
https://developers.facebook.com/docs/whatsapp/cloud-api/webhooks/payload-examples#text-
messages
 if (req.body.object) {
  if (
   req.body.entry &&
   req.body.entry[0].changes &&
   req.body.entry[0].changes[0] &&
   req.body.entry[0].changes[0].value.messages &&
   req.body.entry[0].changes[0].value.messages[0]
  ) {
   let phone_number_id =
    req.body.entry[0].changes[0].value.metadata.phone_number_id;
   let from = req.body.entry[0].changes[0].value.messages[0].from; // extract the phone
number from the webhook payload
   let msg_body = req.body.entry[0].changes[0].value.messages[0].text.body; // extract the
message text from the webhook payload
   axios({
    method: "POST", // Required, HTTP method, a string, e.g. POST, GET
    url:
     "https://graph.facebook.com/v12.0/" +
     phone_number_id +
     "/messages?access_token=" +
     token,
    data: {
     messaging_product: "whatsapp",
     to: from,
     text: { body: "Ack: " + msg_body },
    },
    headers: { "Content-Type": "application/json" },
   });
  }
```

```
    res.sendStatus(200);
  } else {
    // Return a '404 Not Found' if event is not from a WhatsApp API
    res.sendStatus(404);
  }
});

// Accepts GET requests at the /webhook endpoint. You need this URL to setup webhook
initially.
// info on verification request payload: https://developers.facebook.com/docs/graph-
api/webhooks/getting-started#verification-requests
app.get("/webhook", (req, res) => {
  /**
   * UPDATE YOUR VERIFY TOKEN
   *This will be the Verify Token value when you set up webhook
  **/
  const verify_token = process.env.VERIFY_TOKEN;
  console.log(verify_token)
  // Parse params from the webhook verification request
  let mode = req.query["hub.mode"];
  let token = req.query["hub.verify_token"];
  let challenge = req.query["hub.challenge"];

  // Check if a token and mode were sent
  if (mode && token) {
    // Check the mode and token sent are correct
    if (mode === "subscribe" && token === verify_token) {
      // Respond with 200 OK and challenge token from the request
      console.log("WEBHOOK_VERIFIED");
      res.status(200).send(challenge);
    } else {
      // Responds with '403 Forbidden' if verify tokens do not match
      res.sendStatus(403);
    }
  }
});
```

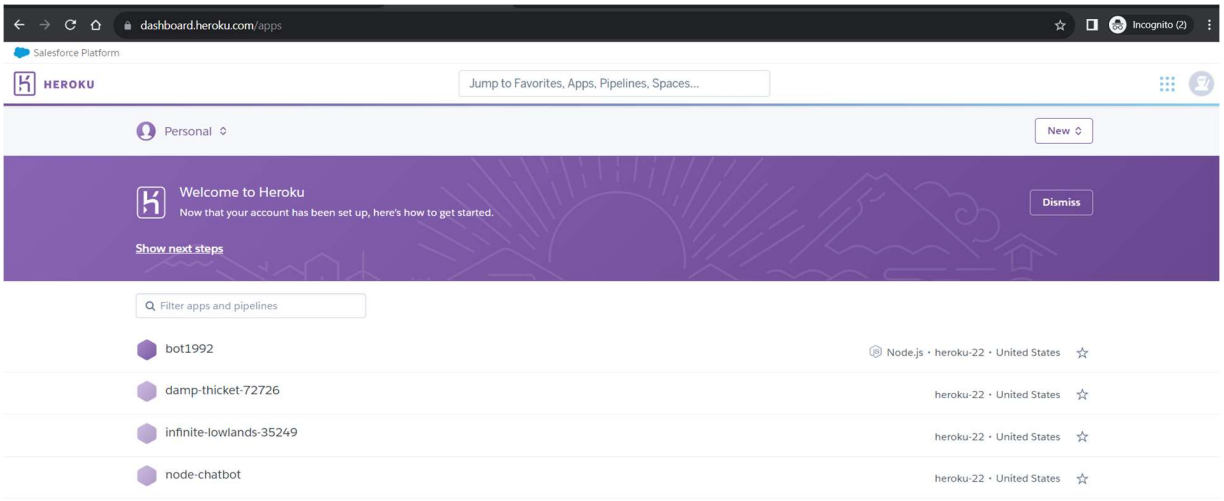
3. Install & configure git

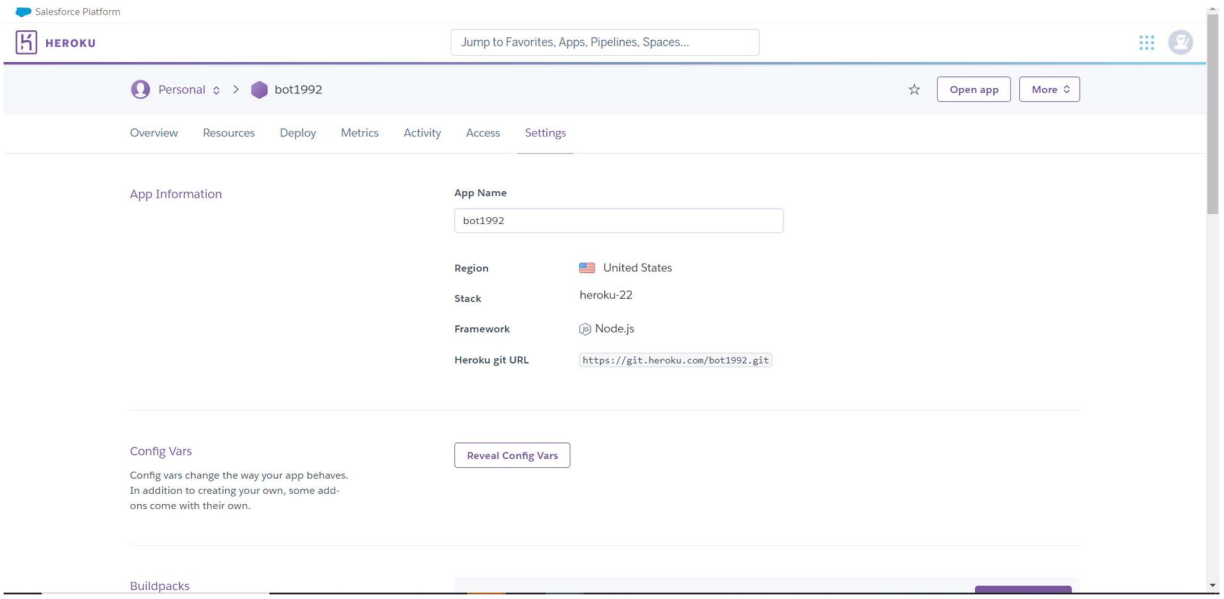
```
ubuntu@ip-<ipaddress>:~/testbot$ git config --list
user.name=git-vignesh
user.email=muthuvignesh92@outlook.com
init.defaultbranch=main
```

4. create a Heroku account,

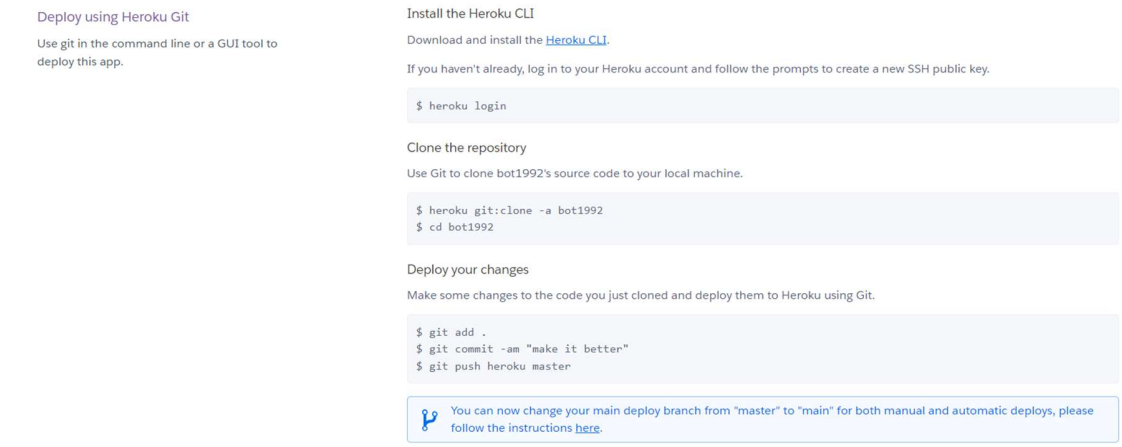
Whatsapp endpoint will only accept https enabled applications, so that we upload the application in Heroku and share the https enpoint in whatsapp business.

Create a application





Steps to update the changes in git and push to Heroku

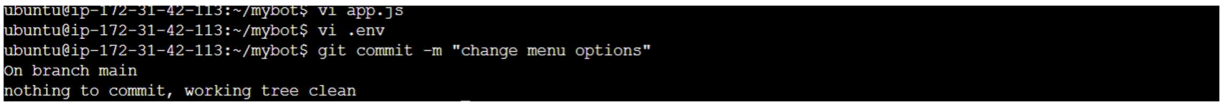


5. Login to machine
Create a .env file inside the project folder

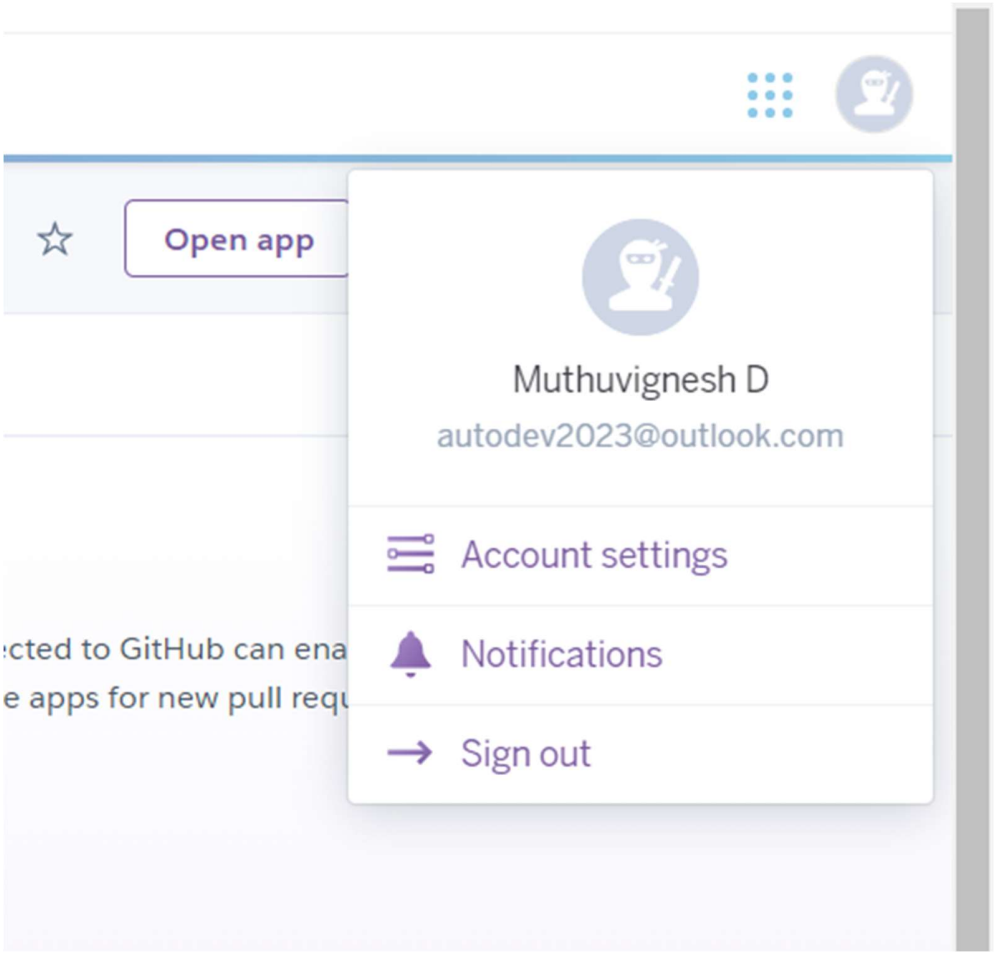
Vi .env

Copy and paste the whastapp token

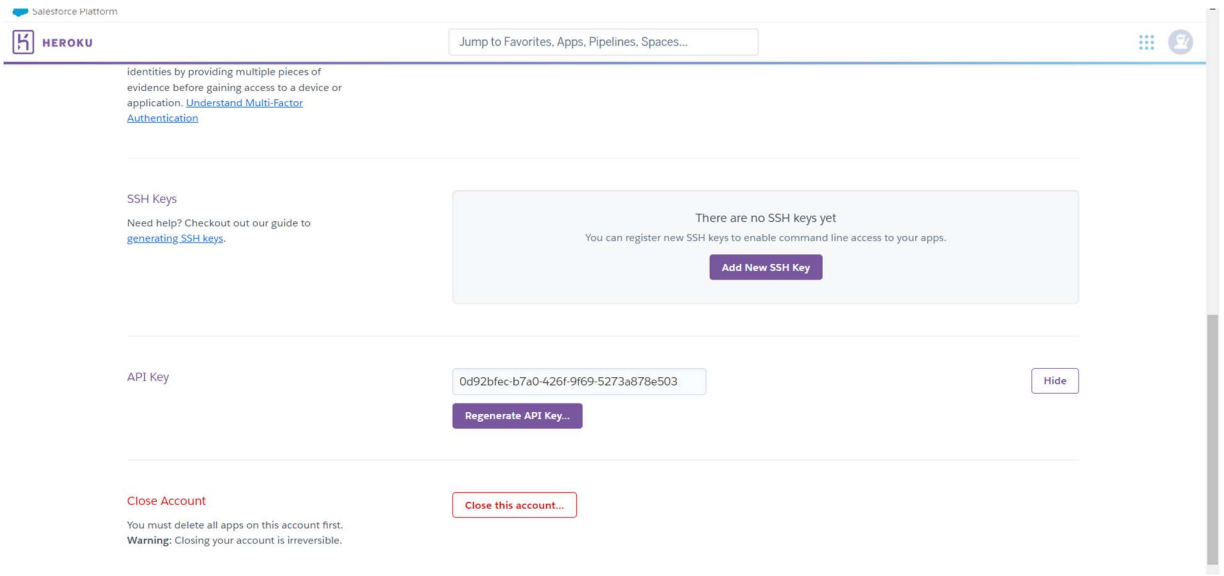
Save the file
Commit the changes



Login to Heroku with username and API KEY
code



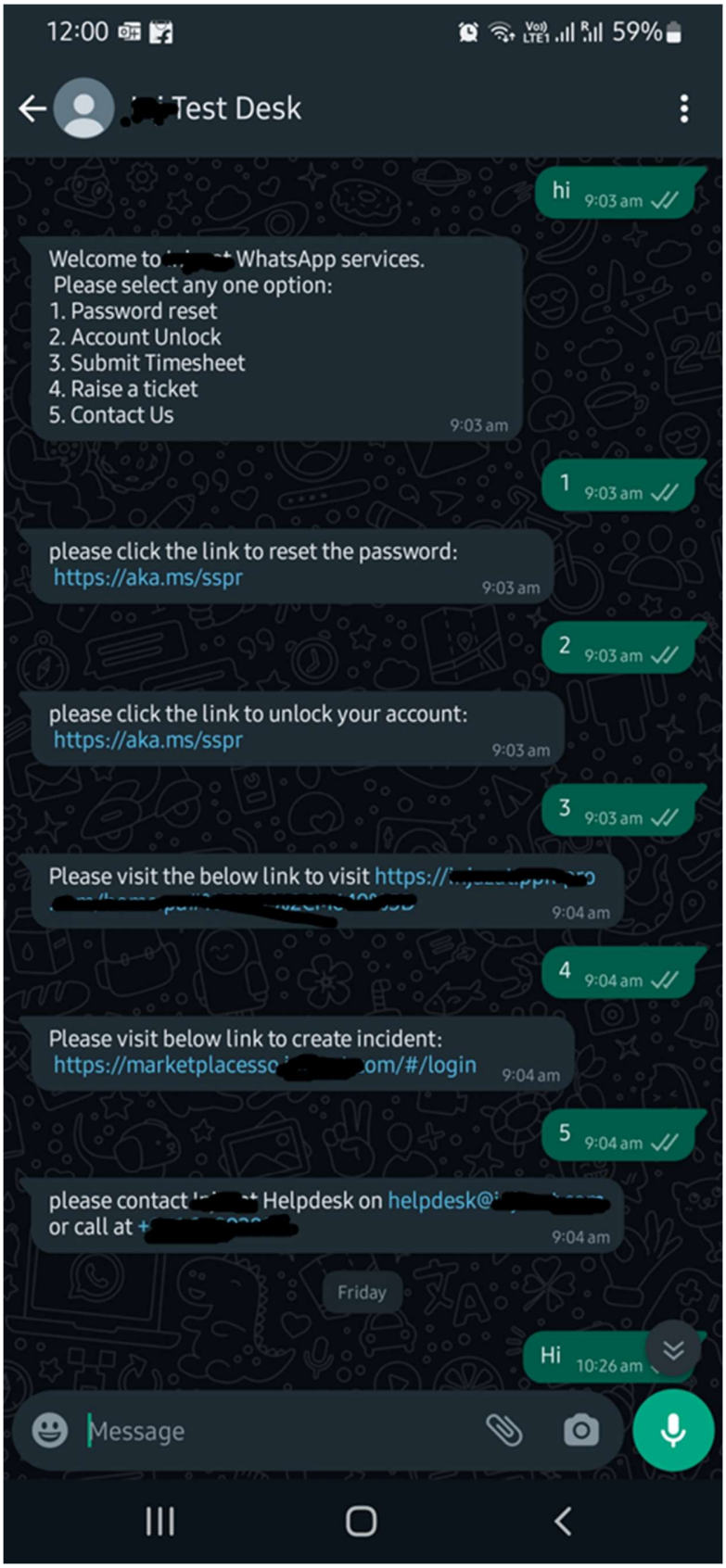
Account settings -> api key



```
ubuntu@ip-172-31-42-113:~/mybot$ heroku login -i
heroku: Enter your login credentials
Email [autodev2023@outlook.com]: autodev2023@outlook.com
Password:
```

```
ubuntu@ip-172-31-42-113:~/mybot$ heroku login -i
heroku: Enter your login credentials
Email [autodev2023@outlook.com]: autodev2023@outlook.com
Password: **********************************
Logged in as autodev2023@outlook.com
ubuntu@ip-172-31-42-113:~/mybot$
```

```
ubuntu@ip-172-31-42-113:~/mybot$ git push heroku main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 12.39 KiB | 4.13 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Updated 1076 paths from 15d567a
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-22 stack
remote: -----> Using buildpack: heroku/nodejs
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NODE_VERBOSE=false
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
remote:
```


12:00
59%
Test Desk
hi 9:03 am
Welcome to WhatsApp services.
Please select any one option:
1. Password reset
2. Account Unlock
3. Submit Timesheet
4. Raise a ticket
5. Contact Us
9:03 am
1 9:03 am
please click the link to reset the password:
https://aka.ms/sspr
9:03 am
2 9:03 am
please click the link to unlock your account:
https://aka.ms/sspr
9:03 am
3 9:03 am
Please visit the below link to visit https://
9:04 am
4 9:04 am
Please visit below link to create incident:
https://marketplacesso om/#/login
9:04 am
5 9:04 am
please contact Helpdesk on helpdesk@
or call at +
9:04 am
Friday
Hi 10:26 am
Message