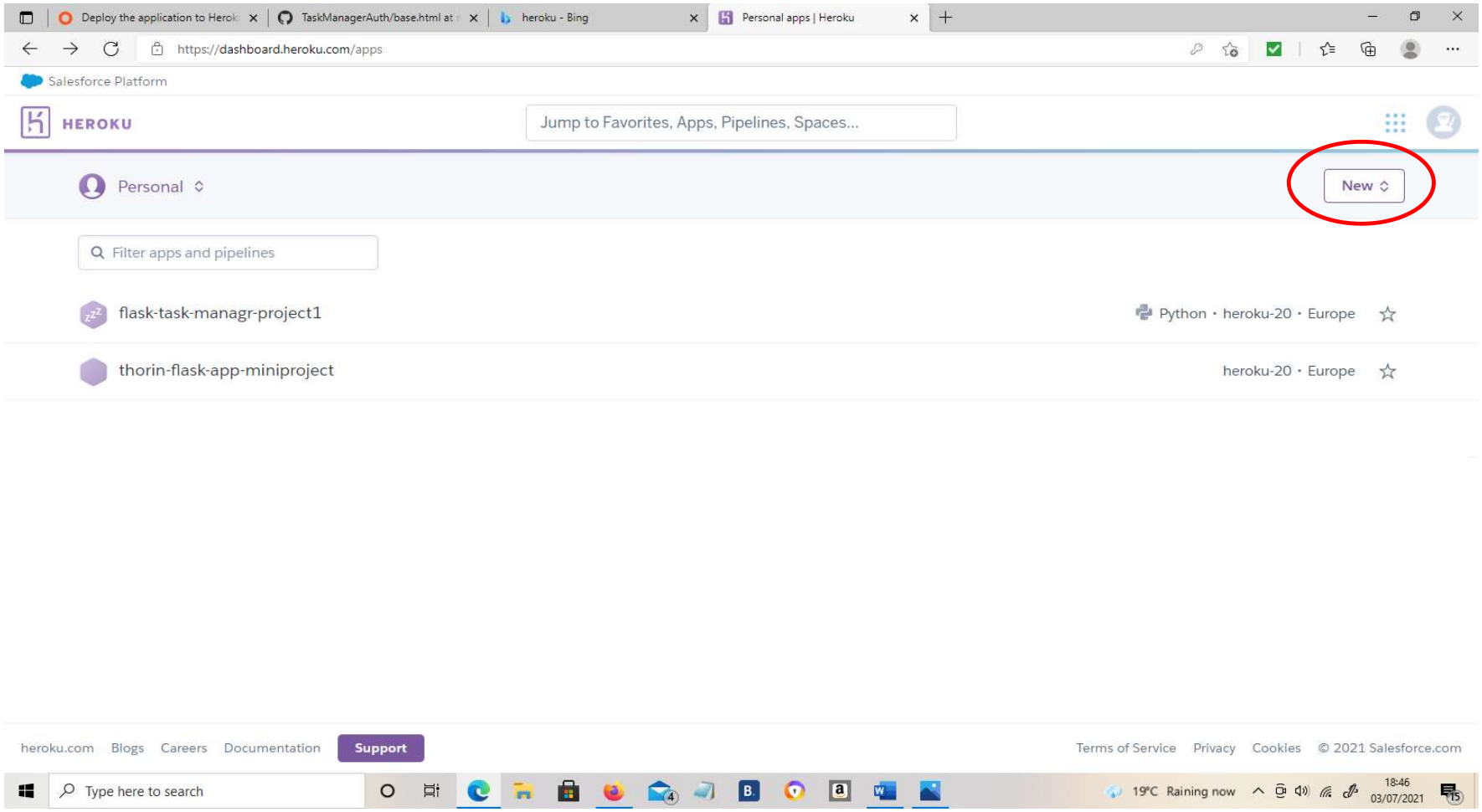
Deployment to Heroku

Deploying to Heroku

There are four steps to deploying to Heroku:-

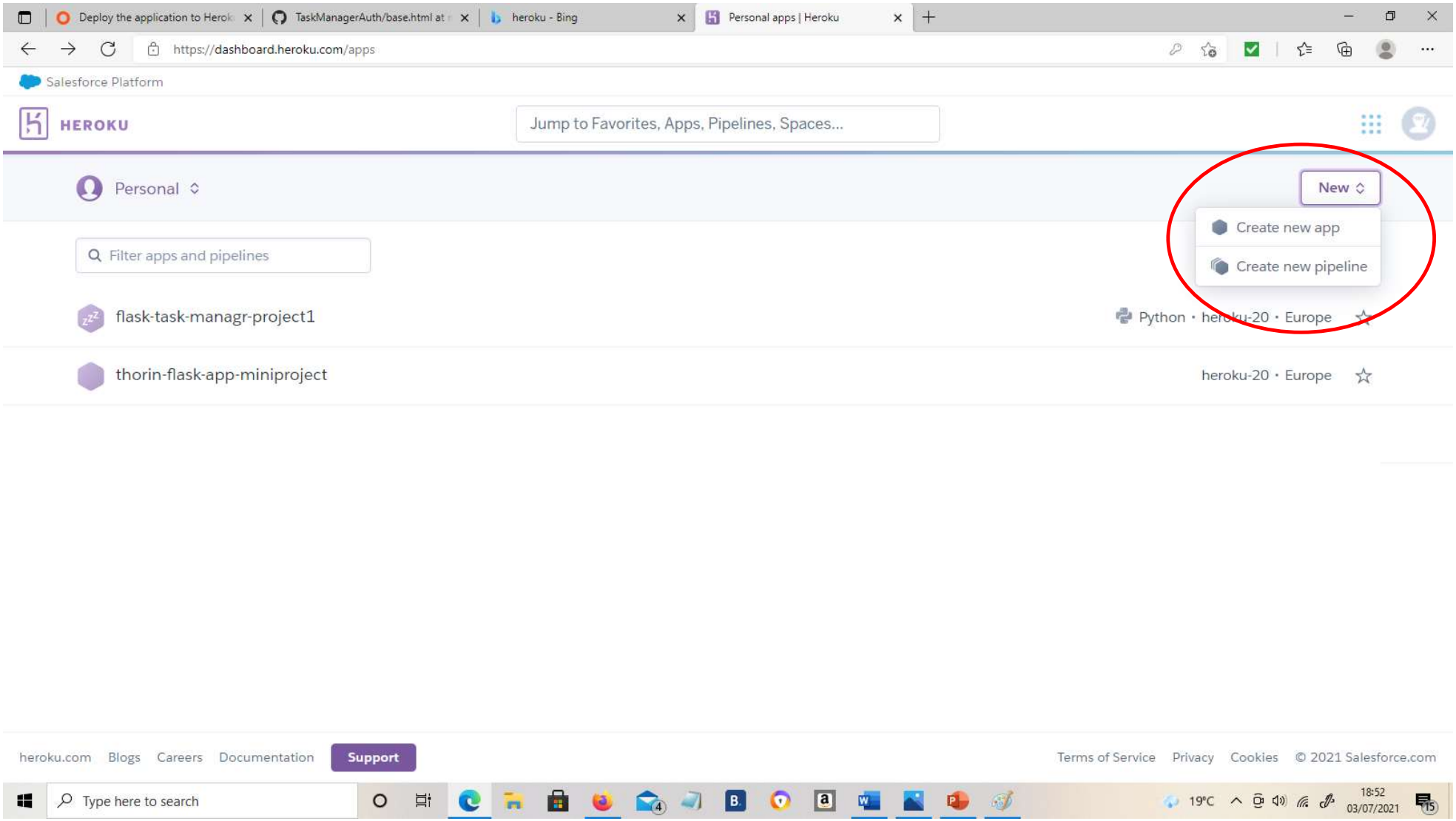
1. Create a Heroku app
2. Link local Git repository with Heroku **or** set up Automatic deployment
3. Create a requirements.txt file which will install dependencies for the project, this tells Heroku that it uses Python.
4. Create the Procfile. A Procfile is a Heroku-specific type of file that tells Heroku how to run the project.

Step 1. Deploying to Heroku – Creating the Heroku app



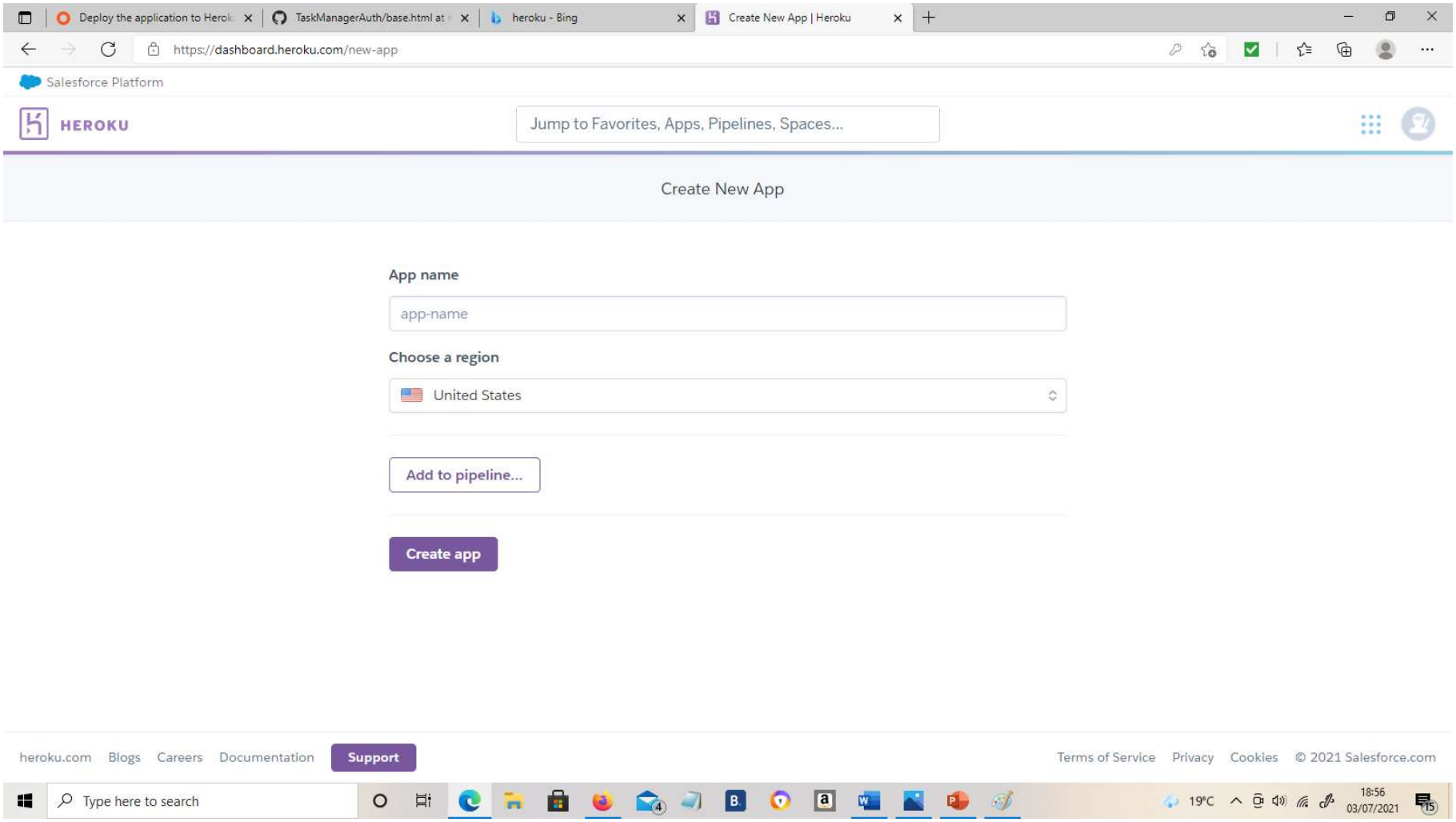
Login to Heroku, click on ‘New’.

Step 1. After clicking ‘New’, click ‘Create new app’.



Step 1. Name the app, choose region, then Create App.

Names must be unique and use all lowercase letters, dashes instead of spaces.



Step 2a. Deployment to Heroku using CLI (Heroku Toolbelt)

At the command line in project (Gitpod)

‘npm install –g Heroku’

Node Package Manager NPM -g means ‘install Heroku globally’

‘heroku login –i’

Next, login to Heroku

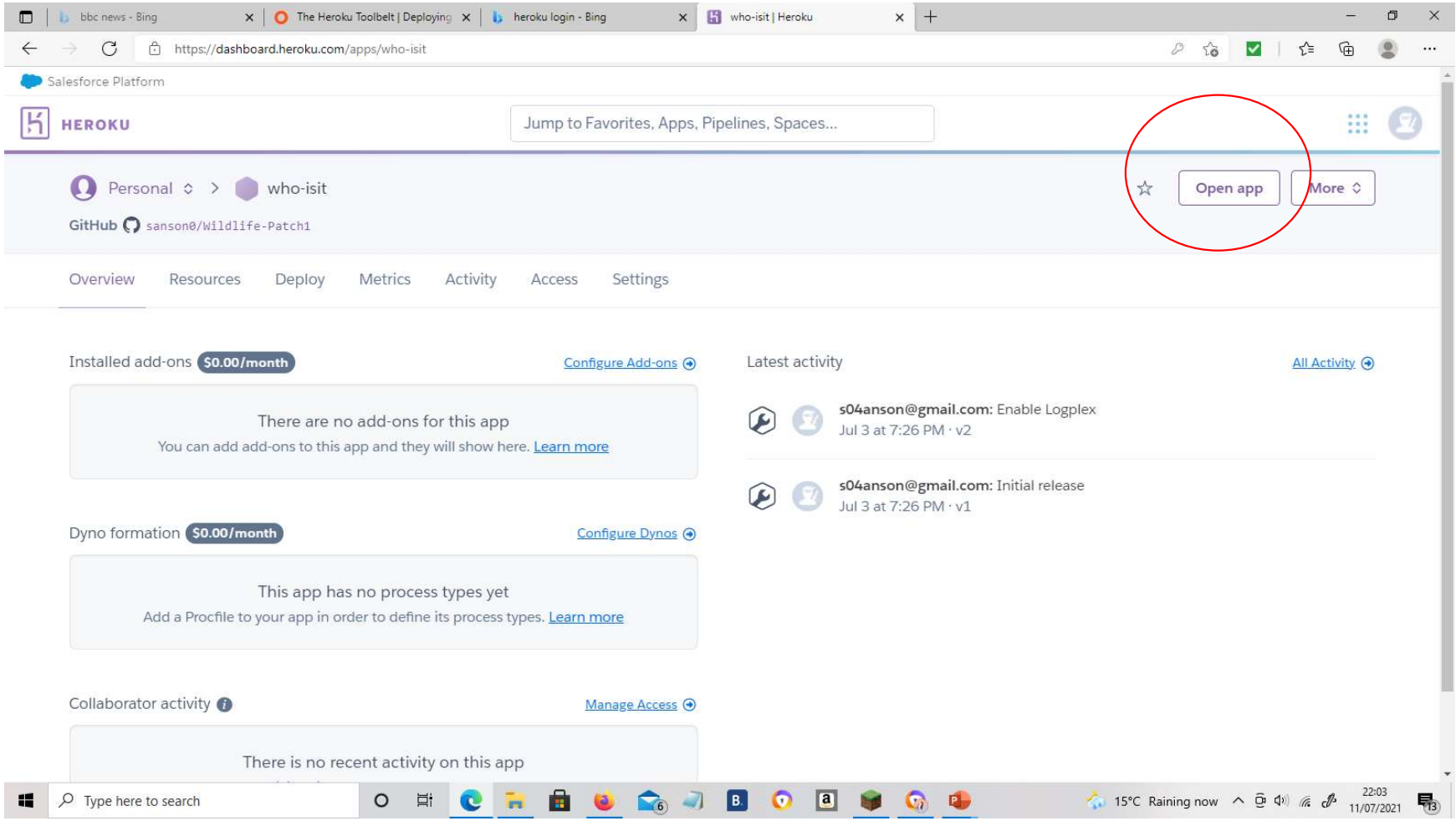
‘Heroku apps’

lists all the apps a user has created

Live site has an address similar to https://Name-of-project.herokuapp.com/

Control –click on the address to launch site or ‘Open app’ in the Heroku site. This is the default Heroku app at the moment but with user’s project name.

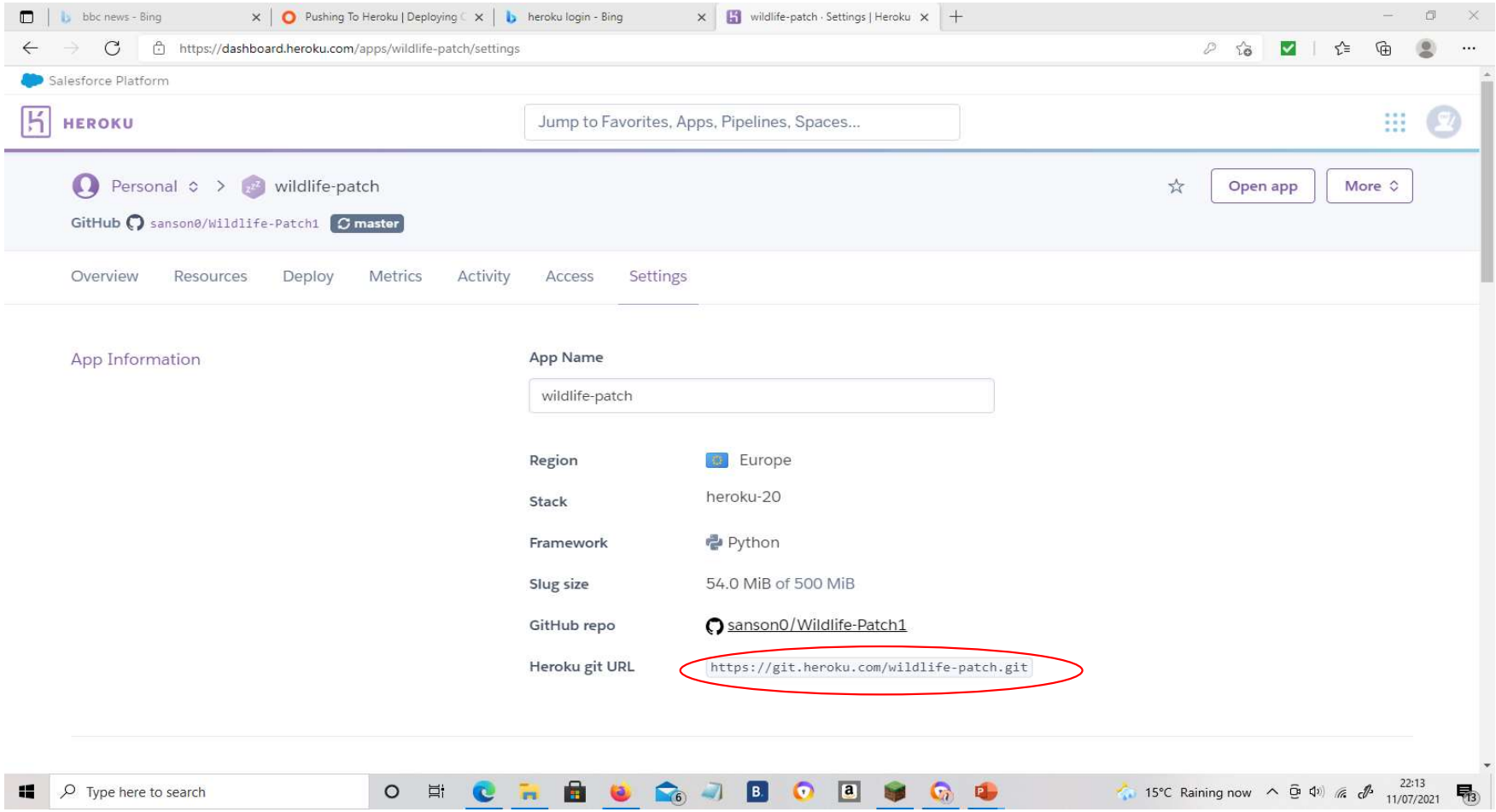
Step 2a. Deployment to Heroku App can be opened using ‘Open app’



Step 2a. Connect Git remote to Heroku

- At the command line type ‘git status’ to check on project files status
- Workspace linked to Github repository? If not type ‘git init’
- Add files to staging area ‘git add –A’
- Then ‘git commit –m “Deployment to Heroku”’

Step 2a. Connect Git remote to Heroku



Copy git url circled in red

Step 2a. Connect Git remote to Heroku

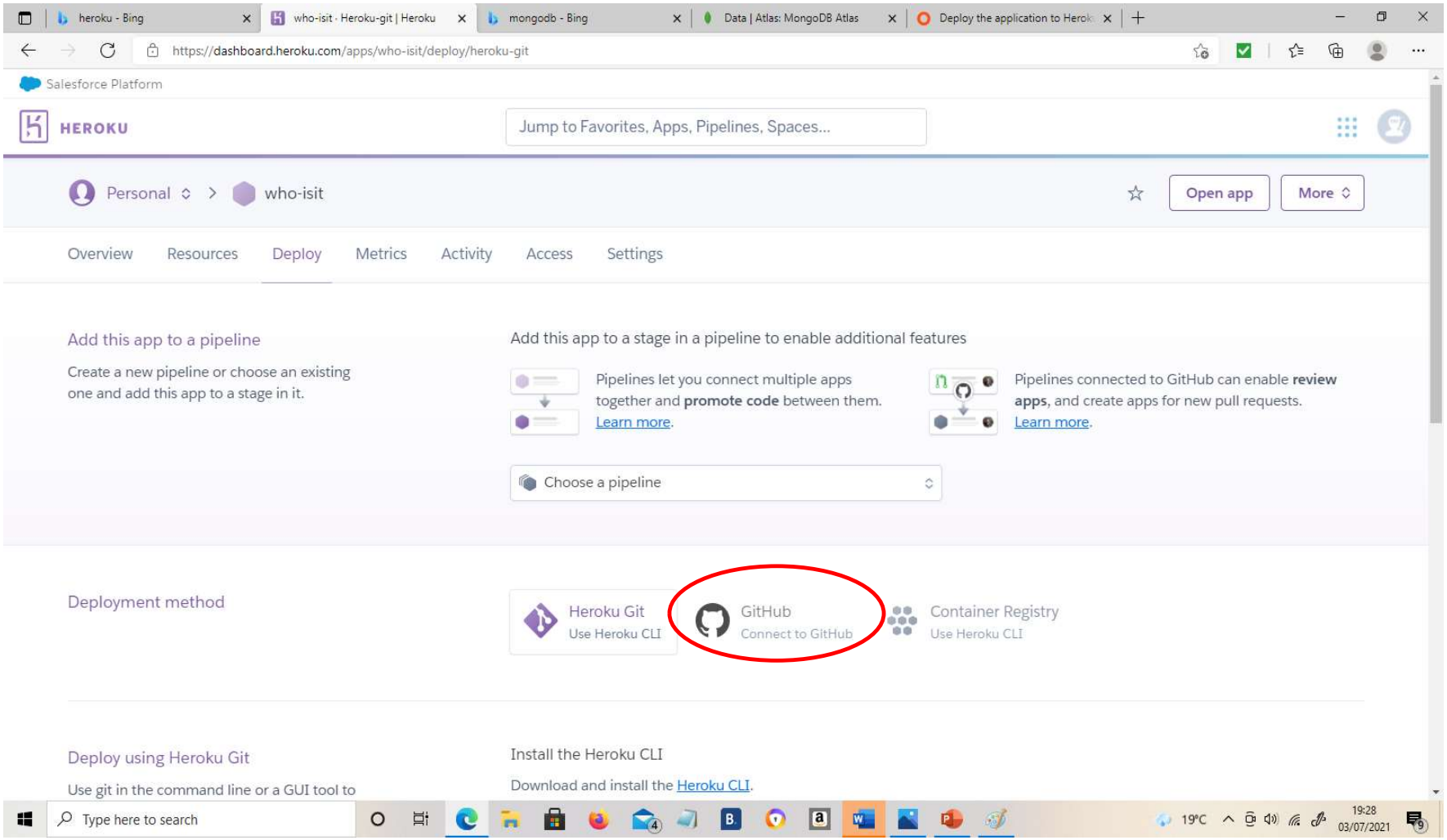
To add another remote, type ‘git remote add’ then paste the Heroku Git URL next to the command, press return
When ‘git remote –v’ is typed press return, the Heroku Git URL and original Github URL are displayed.

To push code directly to Heroku, type ‘git push –u Heroku master’ press return

Currently an error displays “No default language could be detected for this app”

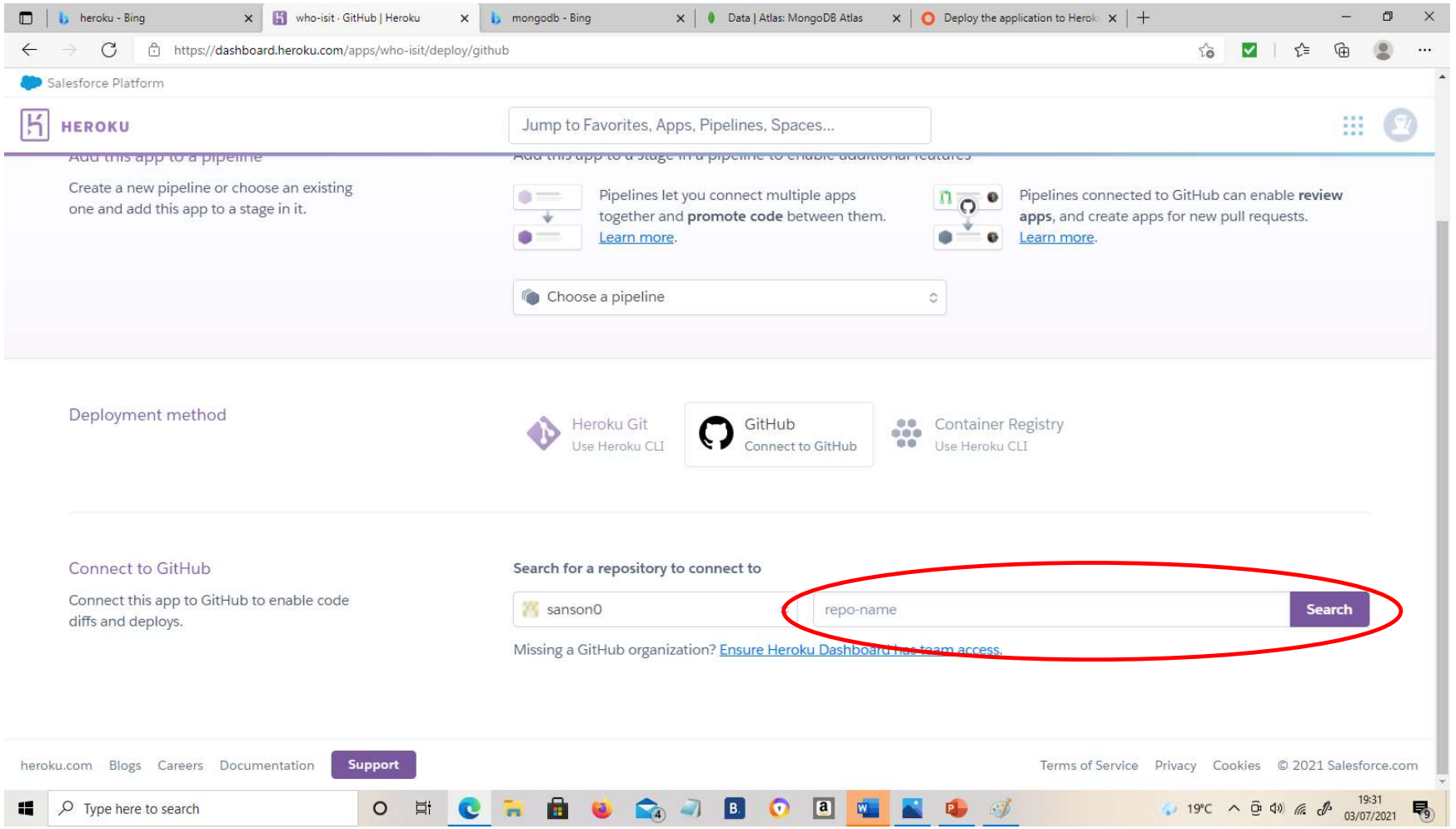
The next step (3) is required to tell Heroku that the language required is Python, this step is to create a requirements.txt file.

Step 2b. Automatic deployment Click on ‘Connect to Github’ under ‘Deploy’ menu



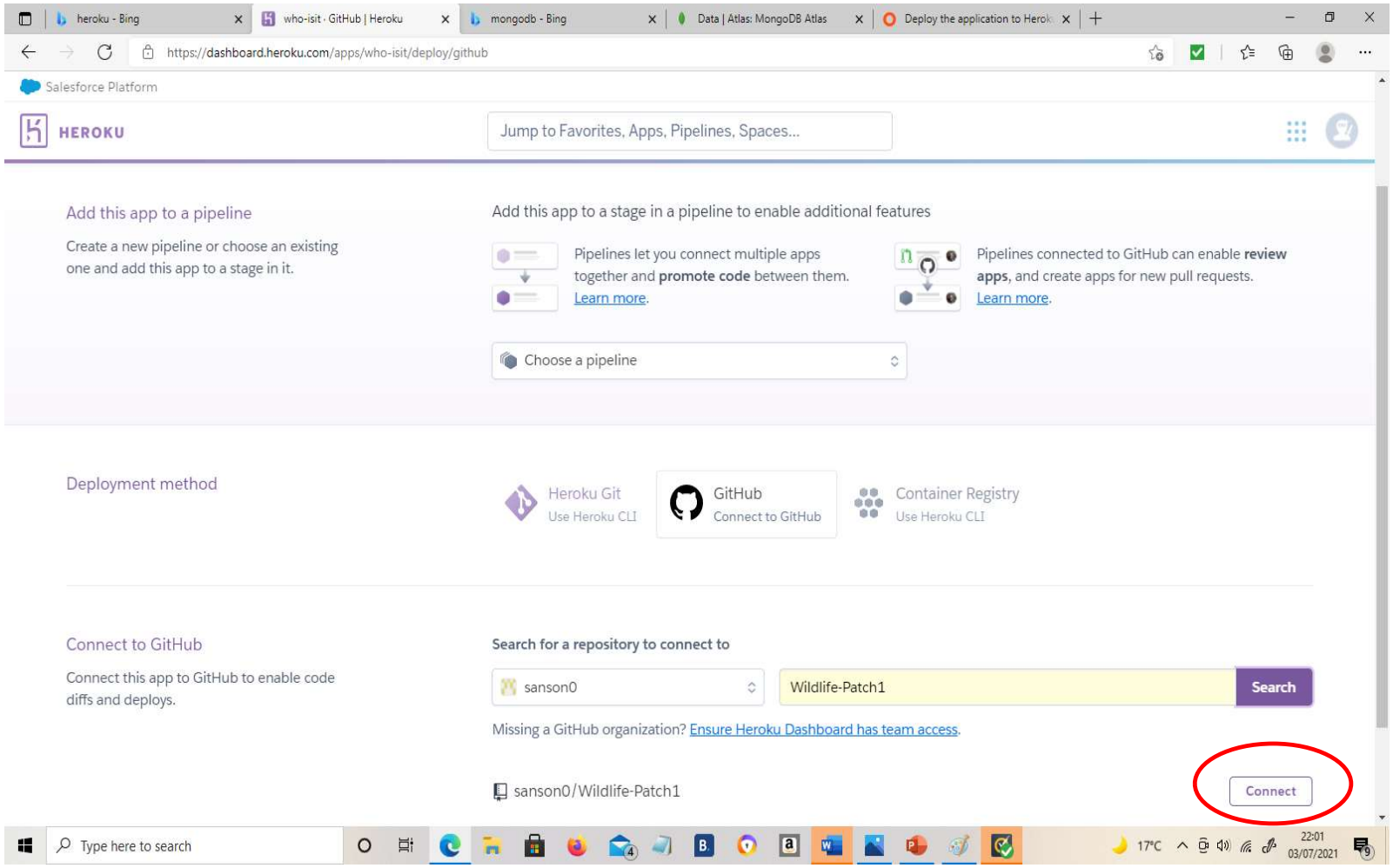
Click on ‘Github’

Step 2b. ‘Connect to Github’ on the Heroku site



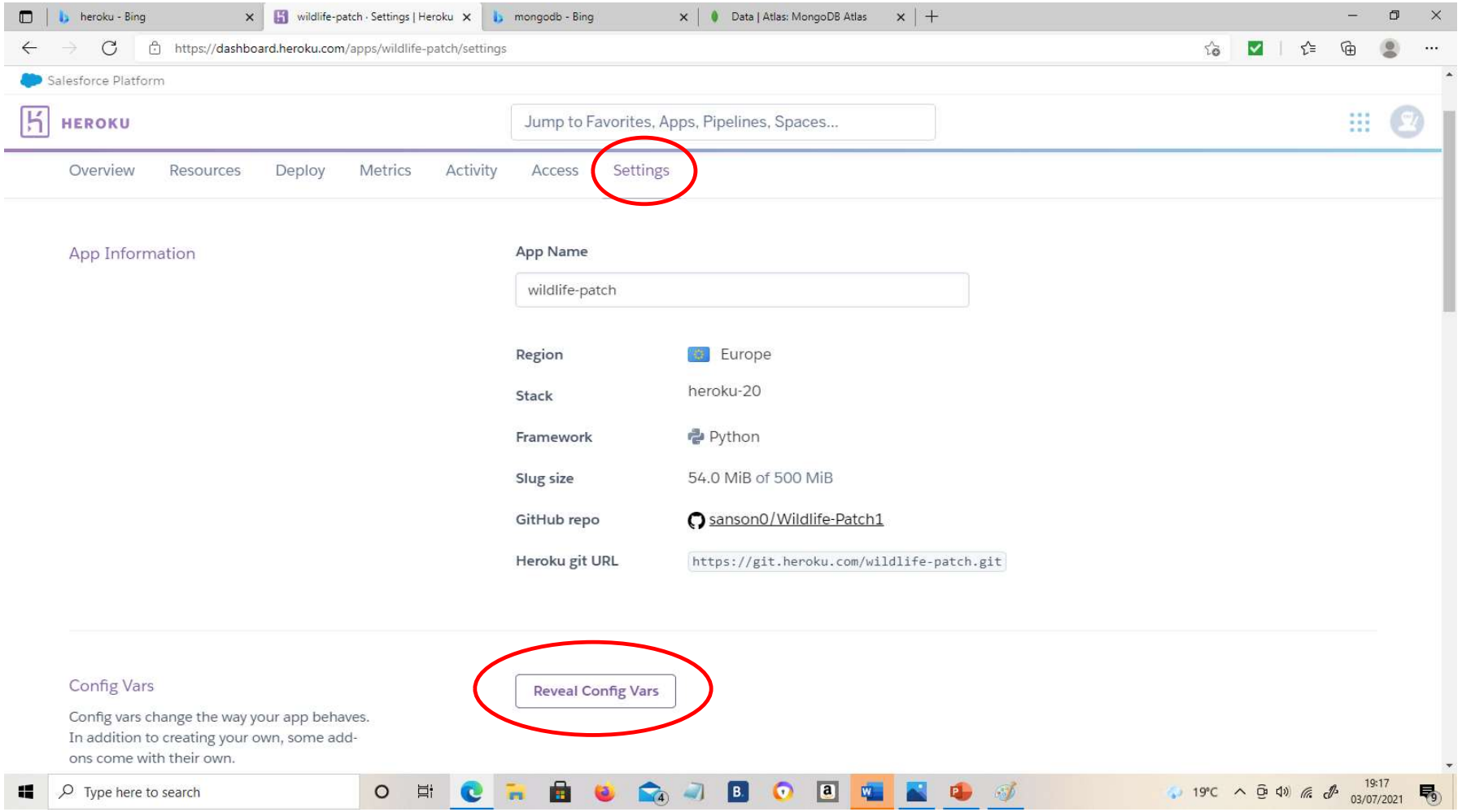
Find repo name

Step 2b. ‘Connect’ after searching repository name



When repo is found click ‘Connect’, don’t click ‘Automatic deployment’ yet…

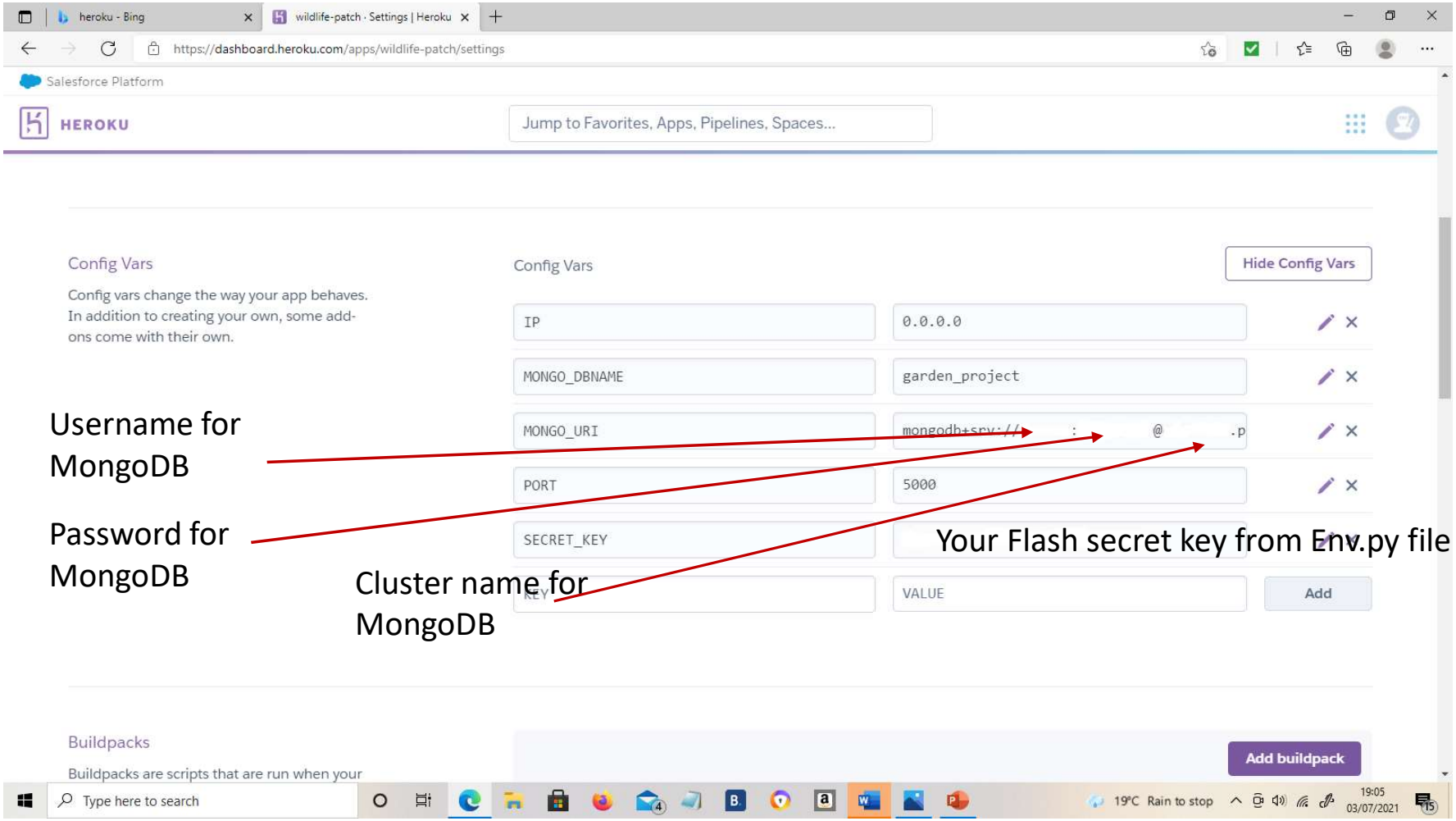
Step 2b. Automatic deployment – after clicking ‘Settings’



Go into ‘Settings’ and click ‘Config Vars’

Step 2b. Automatic deployment – after clicking ‘Config Vars’

Step 2b. Automatic deployment – after clicking ‘Config Vars’



Username for MongoDB

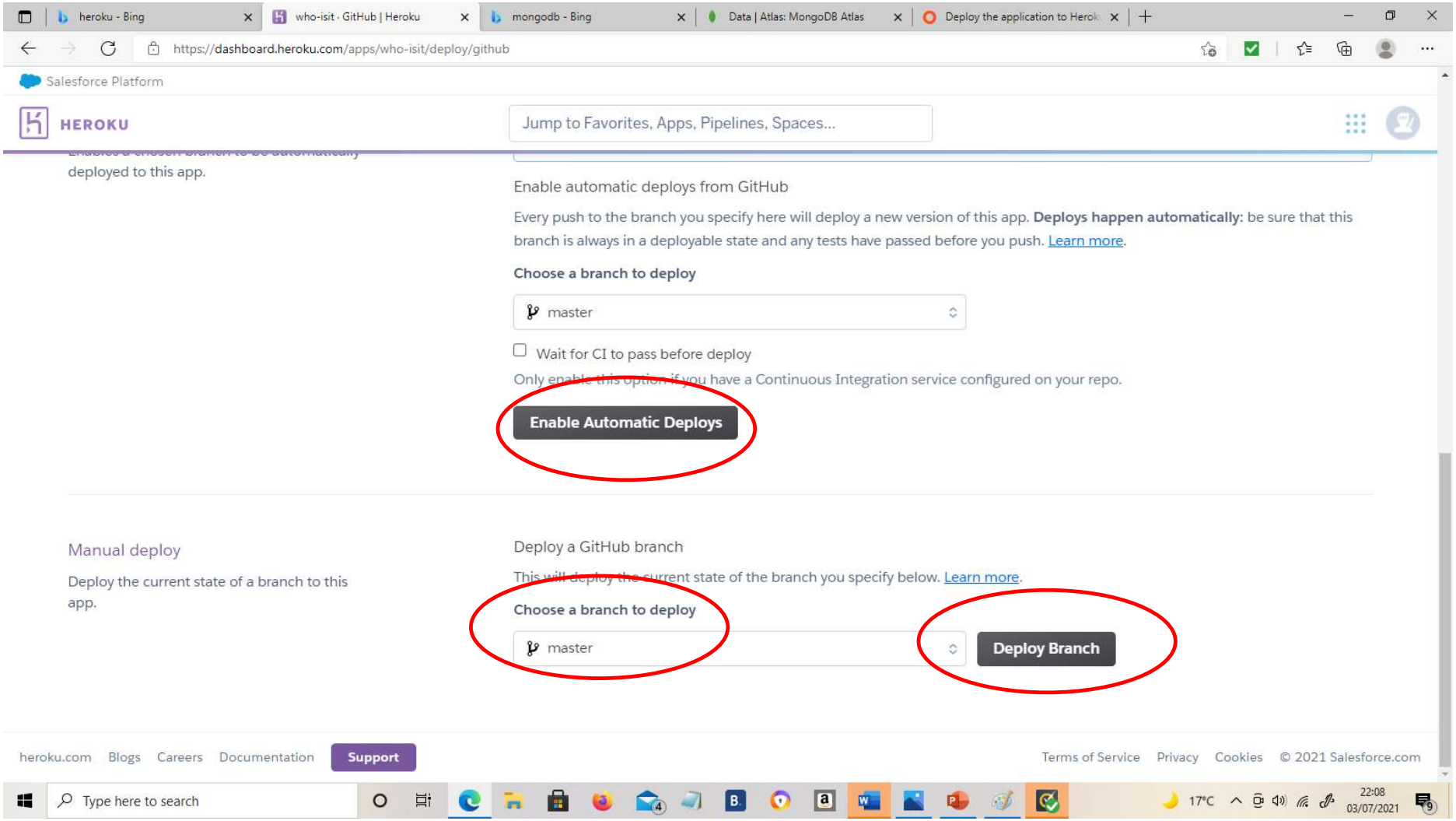
Password for MongoDB

Cluster name for MongoDB

Your Flash secret key from Env.py file

Don’t include quotes “”. Then click ‘Hide Vars’ and go to deploy menu

2b. Automatic deployment – after clicking master branch, deploy branch but WAIT before clicking 'Enable Automatic Deploys'. Steps 3 and 4 are required first.



Step 3. Create requirements.txt file

Type ‘pip3 freeze –local > requirements.txt

Output of freeze command goes to requirements.txt

This tells Heroku which applications and dependencies are required to run the app.

‘git add –A’ (Adds requirements.txt to staging area)

‘git commit – “Add requirements.txt”

‘git push –u heroku master’ (push code to Heroku)

This installs python and pip installed all the other requirements such as Flask, Jinja, Click, etc.

‘Application error’ shows as the Procfile needs to be created.

Step 4. Creating the Procfile

A Procfile is a Heroku-specific type of file that tells Heroku how to run the project

Type ‘echo web:python app.py > Procfile’

‘git add Procfile’

‘git commit –m “Add Procfile”’

‘git push’

Deploy to Heroku

- Make sure that the Procfile and requirements.txt file are pushed to Github before automatic deployment button is clicked.

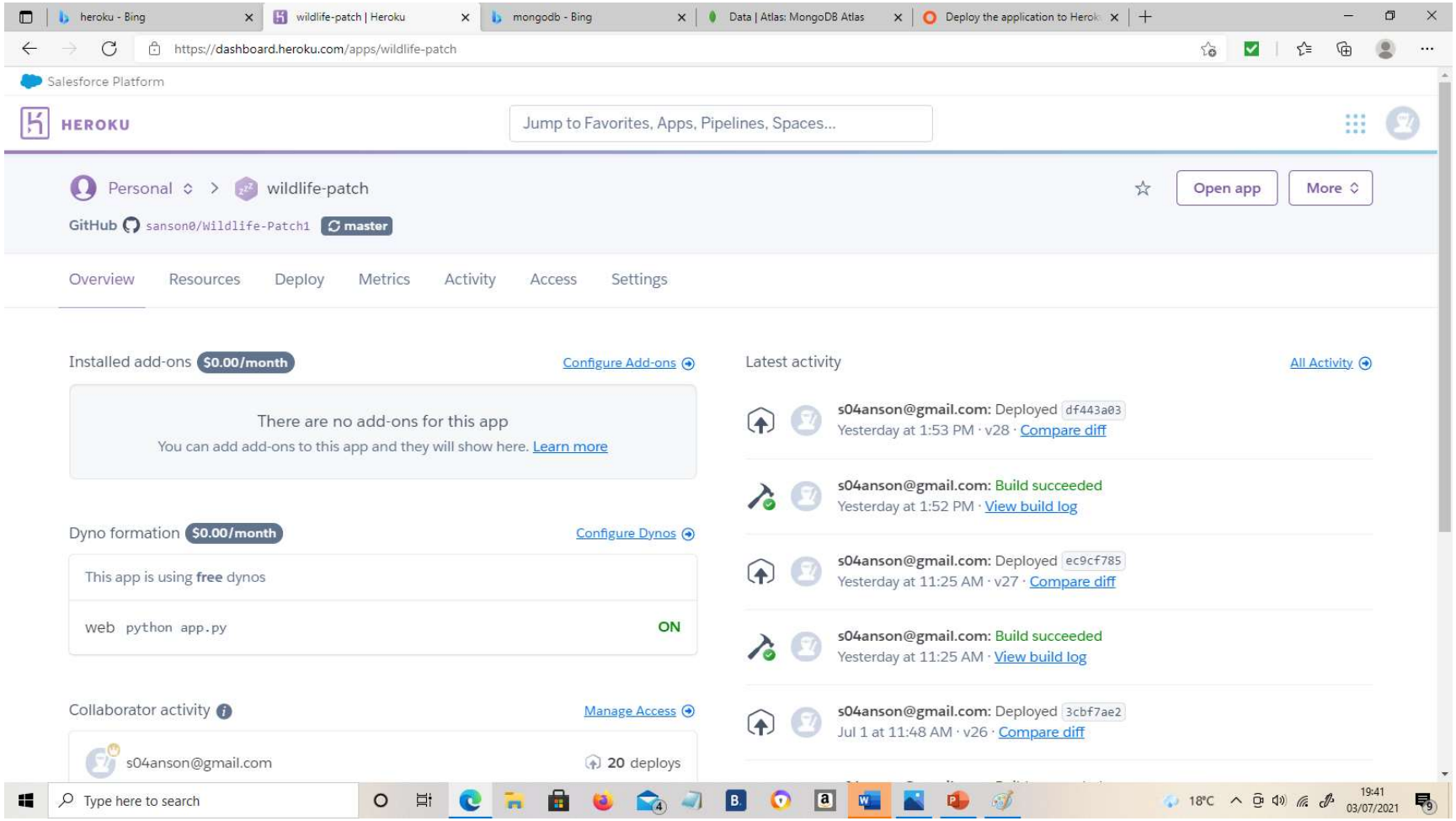
If deploying to Heroku using automatic deployment

Remove the Heroku remotes by typing

‘git remote rm Heroku’

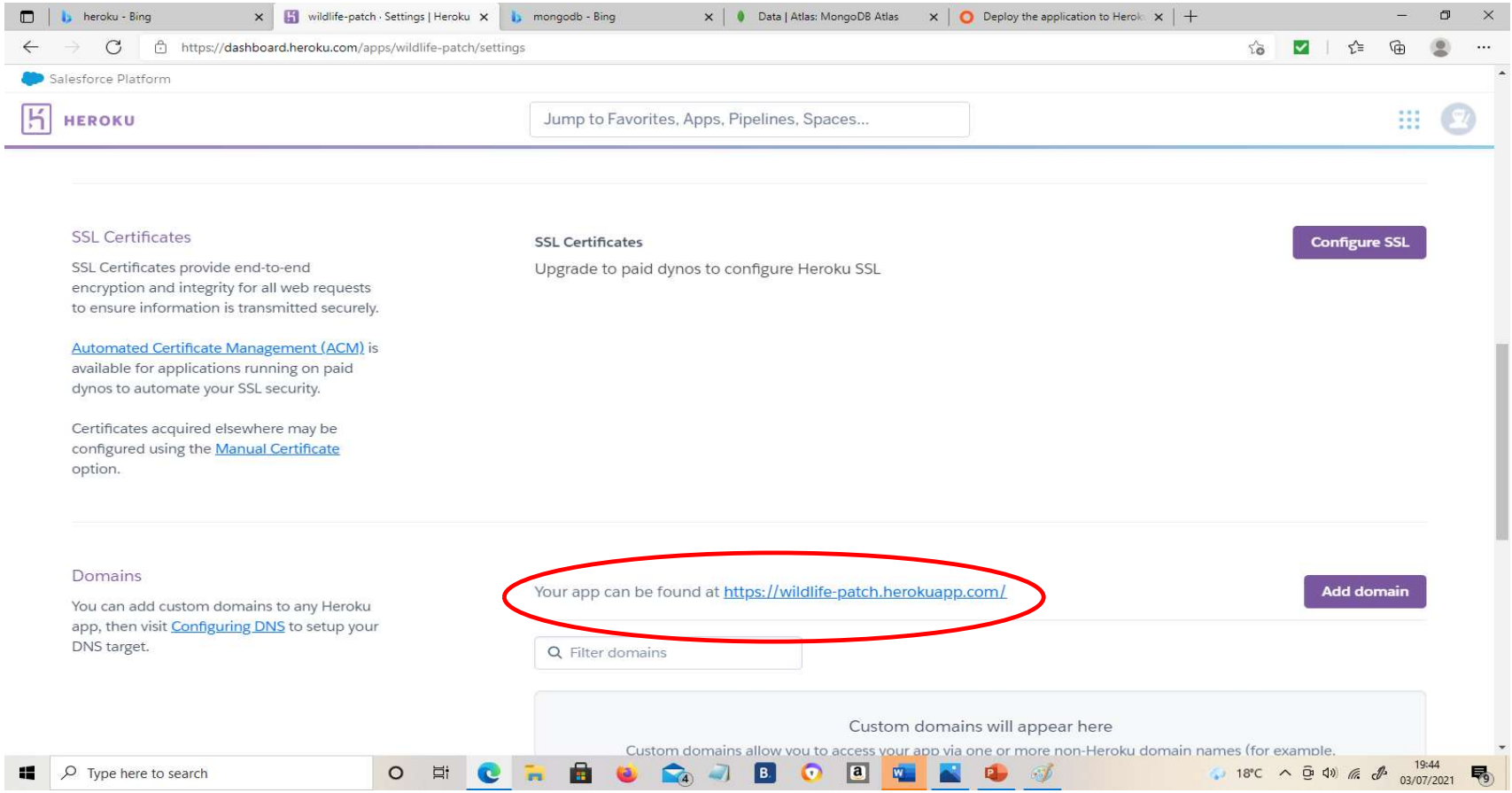
then only Github is connected to the app

Deploy to Heroku



Your app is successfully deployed to Heroku

Deploy to Heroku



After the app is deployed, the url is highlighted by the red ellipse.

Automatic Deploy to Heroku

Once the app is deployed it is updated automatically whenever changes are made to the code.