# HOSTING

# Hosting Services

- At some point we are going to want to provide our website/app/service to other users.  We need a host.
  - *Do it ourself.*
  - *Google Cloud Services*
  - *Amazom AWS*
  - *Microsoft Azure*
  - *Github Pages*
  - *Heroku*
  - *Many others*

# Heroku

- It has a free tier and does not ask for a credit card.
- https://www.heroku.com
- Signup https://signup.heroku.com/dc
  - *Would choose Node.js even though it is not a language.*
- Getting started with node.js https://devcenter.heroku.com/articles/getting-started-with-nodejs
- Heroku requires npm, node.js, and git

# Heroku

- Install the command line interface.  Heroku has platform specific instructions for installation on Windows/MacOS
- I would be surprised if your school laptop was running the 32bit OS for Windows, but we can check. (Windows 8 and 10.  For earlier versions, search online.)
  - *Press Windows key and E*
  - *Right Click This PC*
  - *Select Properties from the menu*
  - *You can see the details in the window that opens.*
- If you have the 64 bit OS, use the corresponding installer.

# Login

- Run from the command line

```
heroku login
```

- Will pop up a browser window where you can login – may persist credentials.
- Also these to verify installations

```
node –version
npm --version
git --version
```

# Locate source

- Navigate to the root directory of your node.js application.
  - *Can be a clone of a Git repository*
  - *Can be a local Git repo attached to a remote. official documentation and download*

# Create the App on Heroku

- From the command line

  heroku create myappname

- This will modify your config file in .git so that there is a remote set up with Heroku.  We will still have the remote for github.
- Push the code up to heroku.

  git push heroku main

# Start it

- From the command line

  heroku ps:scale web=1

- Now that we have an instance running, we can use the URL to check it out or short cut with the following command.

  heroku open

URLs will be something like the following.  Refer back to the Heroku create for the exact URL.

- https://myappname.herokuapp.com/
- https://myappname.herokuapp.com/users

# Check activity

- From the command line
  `heroku logs --tail`

- We can see the requests that have come into the server. A single request triggers other requests.

# Create a Procfile

- We can instruct Heroku how to start the app.  We want to indicate that this is a web app and should be using the HTTP stack and the command to start it up. We can find this in the package.json

- Contents  of Procfile

  *web: npm start*

# Status check

- From the command line

  `heroku ps`

- You can scale up/down the number of responders (dynos)

  `heroku ps: scale web=#`

- Life is happy (free) if there are not too many requests and only 0 or 1 dynos.
- Dynos can go to sleep and restart can be slow.

# Run Local

- From the command line

  `heroku local web`

- It will respond on localhost:5000

  `heroku ps: scale web=#`

- Life is happy (free) if there are not too many requests and only 0 or 1 dynos.
- Dynos can go to sleep and restart can be slow.