**Installing and Upgrading Node-RED**

We provide a script to install Node.js, npm and Node-RED onto a Raspberry Pi. The script can also be used to upgrade an existing install when a new release is available.

Running the following command will download and run the script. If you want to review the contents of the script first, you can view it [here](https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered).

bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

### Running locally

As with [running Node-RED locally](https://nodered.org/docs/getting-started/local), you can use the node-red command to run Node-RED in a terminal. It can then be stopped by pressing Ctrl-C or by closing the terminal window.

Due to the limited memory of the Raspberry Pi, you will need to start Node-RED with an additional argument to tell the underlying Node.js process to free up unused memory sooner than it would otherwise.

To do this, you should use the alternative node-red-pi command and pass in the max-old-space-size argument.

node-red-pi --max-old-space-size=256

**Opening the editor**

Once Node-RED is running you can access the editor in a browser.

If you are using the browser on the Pi desktop, you can open the address: [http://localhost:1880](http://localhost:1880/).

When browsing from another machine you should use the hostname or IP-address of the Pi: http://<hostname>:1880. You can find the IP address by running hostname -I on the Pi.

# **Cloning a repository**

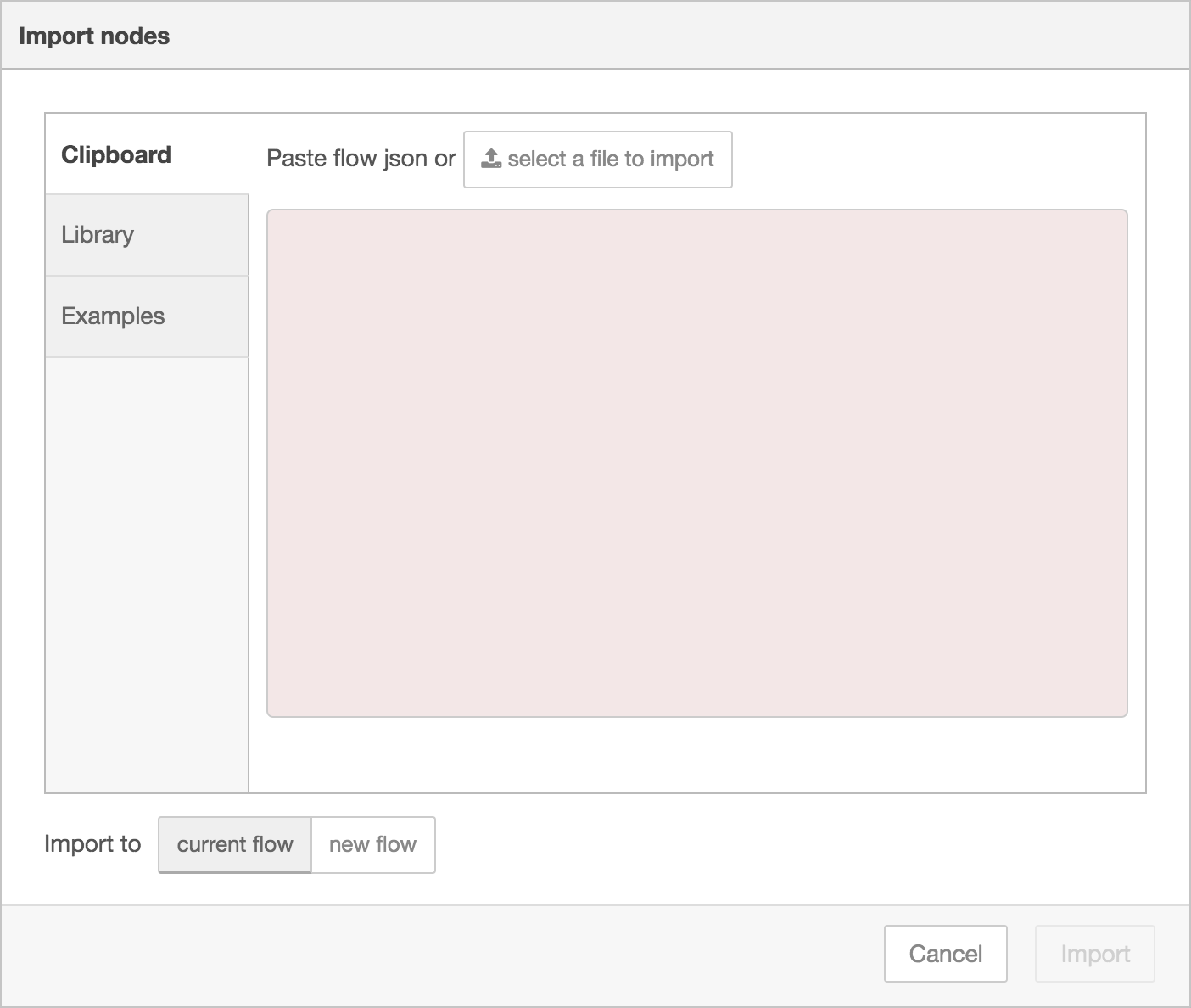
When you create a repository on GitHub, it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.

1. Open Git Bash.
2. Change the current working directory to the location where you want the cloned directory.
3. Type git clone, and then paste the URL you copied earlier.

$ git clone https://github.com/shannenang/PPET2020S1.git

1. Press **Enter** to create your local clone.
2. Navigate to the IoT folder in PPET2020S1 (PPET2020S1/ IoT/Team\_PPET IoT Source Code.json)

**Importing flows**



The Import dialog can be used to import a flow by the following methods:

* Pasting in the flow JSON directly, To import it into the editor, copy it(Team\_PPET IoT Source Code.json) to your clipboard and then paste it into the Import dialog.

|  |
| --- |
| [{"id":"47cc8aac.16df14","type":"function","z":"5e7fc2ab.6deb6c","name":"Healthy or Unhealthy Poop ?","func":"var PoopHealthOutcome = {};\nlet PoopHealthProbability = Number(msg.payload.predictions[0].probability);\n\nif (PoopHealthProbability > 0.5) { //Healthy\n PoopHealthOutcome.payload = \"Poop is Healthy\"\n} \nelse {\n PoopHealthOutcome.payload = \"Poop is Un-healthy\";\n}\n\nreturn PoopHealthOutcome;\n\n","outputs":1,"noerr":0,"x":2541,"y":737,"wires":[["687ac493.07227c","60ff7a9e.5909b4"]]}] |

* Uploading a flow JSON file, (PPET2020S1/ IoT/Team\_PPET IoT Source Code.json)