Running Node-RED locally

Prerequisites

To install Node-RED locally you will need a supported version of Node.js.

If you are on a Raspberry Pi or any Debian-based operating system, including Ubuntu and DietPi, you can use the Pi install script available here.

If you are on an RPM-based operating system, including RedHat, Fedora and CentOS, you can use the RPM install script available here.

Installing with npm

To install Node-RED you can use the `npm` command that comes with node.js:

```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`. More information about installing Node-RED on Windows can be found here.

This command will install Node-RED as a global module along with its dependencies.

You can confirm it has succeeded if the end of the command output looks similar to:

```
+ node-red@1.0.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

Installing with docker

To run in Docker in its simplest form just run:

```
docker run -it -p 1880:1880 --name mynodered nodered/node-red
```

For more detailed information see our docker guide.

Installing with snap

If your OS supports Snap you can install Node-RED with:

```
sudo snap install node-red
```

When installed as a Snap package, it will run in a secure container that does not have access to some extra facilities that may be needed for you to use, such as:

- access to main system storage. Can only read/write to local home directories.
- `gcc` - needed to compile any binary components of nodes you want to install
- `git` - needed if you want to use the Projects feature
- direct access to gpio hardware
- access to any external commands your flows want to use with the Exec node (for example).

You can run it in "classic" mode which reduces the container security but then does provide wider access.

Running

Once installed as a global module you can use the `node-red` command to start Node-RED in your terminal. You can use `Ctrl-C` or close the terminal window to stop Node-RED.

```
$ node-red

Welcome to Node-RED
===================

11 Oct 23:43:39 - [info] Node-RED version: v1.0.2
11 Oct 23:43:39 - [info] Node.js  version: v10.16.3
11 Oct 23:43:39 - [info] Darwin 18.7.0 x64 LE
11 Oct 23:43:39 - [info] Loading palette nodes
11 Oct 23:43:44 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
11 Oct 23:43:44 - [info] Settings file  : /Users/nol/.node-red/settings.js
11 Oct 23:43:44 - [info] HTTP Static    : /Users/nol/node-red/web
11 Oct 23:43:44 - [info] Context store  : 'default' [module=localfilesystem]
11 Oct 23:43:44 - [info] User directory : /Users/nol/.node-red
11 Oct 23:43:44 - [warn] Projects disabled : set editorTheme.projects.enabled=true to enable
11 Oct 23:43:44 - [info] Creating new flows file : flows_noltop.json
11 Oct 23:43:44 - [info] Starting flows
11 Oct 23:43:44 - [info] Started flows
11 Oct 23:43:44 - [info] Server now running at http://127.0.0.1:1880/red/
```

You can then access the Node-RED editor by pointing your browser at http://localhost:1880.

The log output provides you various pieces of information:

- The versions of Node-RED and Node.js
- Any errors hit when it tried to load the palette nodes
- The location of your Settings file and User Directory
- The name of the flows file it is using.

Node-RED uses `flows_<hostname>.json` as the default flows file. You can change this by providing the flow file name as argument to the `node-red` command.

Command-line Usage

Node-RED can be started using the command `node-red`. This command can take various arguments:

```
node-red [-v] [-?] [--port PORT] [--safe] [--settings settings.js]
         [--title TITLE] [--userDir DIR] [flows.json|projectName]
```

Option	Description
`-p, --port PORT`	Sets the TCP port the runtime listens on. Default: `1880`
`--safe`	Starts Node-RED without starting the flows. This allows you to open the flows in the editor and make changes without the flows running. When you deploy your changes, the flows are then started.
`-s, --settings FILE`	Sets the settings file to use. Default: `settings.js` in `userDir`
`--title TITLE`	Set process window title
`-u, --userDir DIR`	Sets the user directory to use. Default: `~/.node-red`
`-v`	Enables verbose output
`-?, --help`	Shows command-line usage help and exits
`flows.json|projectName`	If the Projects feature is not enabled, this sets the flow file you want to work with. If the Projects feature is enabled, this identifies which project should be started.

Node-RED uses `flows_<hostname>.json` as the default flows file. If the computer you are running on may change its hostname, then you should ensure you provide a static file name; either as a command-line argument or using the `flowsFile` option in your settings file.

Passing arguments to the underlying Node.js process

There are occasions when it is necessary to pass arguments to the underlying Node.js process. For example, when running on devices like the Raspberry Pi or BeagleBone Black that have a constrained amount of memory.

To do this, you must use the `node-red-pi` start script in place of `node-red`. *Note*: this script is not available on Windows.

Alternatively, if are running Node-RED using the `node` command, you must provide arguments for the node process before specifying `red.js` and the arguments you want passed to Node-RED itself.

The following two commands show these two approaches:

```
node-red-pi --max-old-space-size=128 --userDir /home/user/node-red-data/
node --max-old-space-size=128 red.js --userDir /home/user/node-red-data/
```

Upgrading Node-RED

If you installed Node-RED using the Pi script, you can rerun it to upgrade. The script is available here.

If you have installed Node-RED as a global npm package, you can upgrade to the latest version with the following command:

```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`.

Next steps

- Learn how to secure your editor
- Create your first flow
- Starting Node-RED on boot

Node-RED: Low-code programming for event-driven applications.

A project of the OpenJS Foundation.

- GitHub
- npm
- Documentation
- APIs

- Flow Library
- About
- Code of Conduct
- Community

- Blog
- Twitter
- Forum
- Slack