# Installation and Configuration of Node-RED on Ubuntu 16.04 LTS

## References

1. <https://diyprojects.io/node-red-installation-configuration-ubuntu-16-04-lts/#.WkmDhjdOmUk>
2. <https://www.digitalocean.com/community/tutorials/how-to-connect-your-internet-of-things-with-node-red-on-ubuntu-16-04>
3. <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04>

## Prerequisites

1. Install Ubuntu (Desktop or Server) 16.04 LTS 64-bit and helper tools.

## Install Procedure

1. Open a terminal session and install Node.js with Node Package Manager (npm)

sudo apt-get install curl

curl -sL https://deb.nodesource.com/setup\_8.x | sudo -E bash -

sudo apt-get update

sudo apt-get install nodejs

1. Verify Node.js and Node Package Manager (npm) versions

node -v

* 1. Output should be: v8.9.4

npm -v

* 1. Output should be: 5.6.0

1. Create a user account to run Node-RED.

sudo adduser nodered-daemon

* 1. Set the password to: nodered
  2. Set the Full Name field to: Node-Red Daemon
  3. Set the Room Number field to: (Blank)
  4. Set the Work Phone field to: (Blank)
  5. Set the Home Phone field to: (Blank)
  6. Set the Other field to: (Blank)

sudo usermod -aG sudo nodered-daemon

1. Install Node-RED

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

1. Set <http://localhost:1880> as the home page

sudo sed -i '/user\_pref("browser.startup.homepage\_override.buildID", "[0-9]\+");/ i user\_pref("browser.startup.homepage", "http://localhost:1880/");' /home/$USER/.mozilla/firefox/\*.default/prefs.js

1. Manually run Node-RED to verify it works (Optional)
   1. To run Node-RED, execute the following command: node-red
   2. Open a browser and navigate to: http://localhost:1880
   3. To stop Node-RED, press: Ctrl + C
2. Setup to run as a service
   1. Create the service definition file
      1. Option 1: Download the file from Github & push it to the right directory

sudo wget -P /etc/systemd/system/ https://raw.githubusercontent.com/davidmgoodman/linux-notes/master/install-files/node-red.service

* + 1. Option 2: Manually create it from scratch
       1. Open an empty file named node-red.service

sudo nano /etc/systemd/system/node-red.service

* + - 1. Enter the following text:

[Unit]

Description=Node-RED

After=syslog.target network.target

[Service]

ExecStart=/usr/bin/node-red-pi --max-old-space-size=128 -v

Restart=on-failure

KillSignal=SIGINT

# log output to syslog as 'node-red'

SyslogIdentifier=node-red

StandardOutput=syslog

# non-root user to run as

WorkingDirectory=/home/nodered-daemon/

User= nodered-daemon

Group= nodered-daemon

[Install]

WantedBy=multi-user.target

* + - 1. Save and close by pressing ‘Ctrl+X’ then ‘Y’
  1. Enable the service

sudo systemctl enable node-red

1. Verify the base Node-RED install works as a service
   1. Reboot
   2. Open a browser and navigate to: <http://localhost:1880>
   3. Verify Node-RED is operational
   4. Open a terminal session and run following command: sudo nautilus /home/nodered-daemon/.node-red/
   5. Verify the Node-RED files exist (settings.js, flows\_$MACHINENAME.json)
2. Install useful packages for Node-RED
   1. Procedure to install a Node.js package
      1. Open a terminal session and navigate to the Node-RED directory.

cd /home/nodered-daemon/.node-red/

* + 1. Then run the following command

sudo npm install <npm-package-name>

* 1. Packages to install
     1. node-red-dashboard

sudo npm install node-red-dashboard

* + 1. node-red-contrib-alasql

sudo npm install --unsafe-perm node-red-contrib-alasql

* + 1. node-red-contrib-actionflows

sudo npm install node-red-contrib-actionflows

* + 1. fs-extra

sudo npm install fs-extra

* + 1. jszip

sudo npm install jszip

* + 1. lodash

sudo npm install --save lodash

* + 1. uuid

sudo npm install uuid

* + 1. xmldom

sudo npm install xmldom

* + 1. docxtemplater

sudo npm install docxtemplater

* + 1. deasync

sudo npm install deasync

* + 1. async

sudo npm install async

* + 1. node-opcua

sudo npm install --unsafe-perm node-opcua

* + 1. honcho / nodepccc / nodes7 / mbtcpprotocol

sudo npm install honcho

* + 1. node-red-contrib-modbus

sudo npm install --unsafe-perm --build-from-source node-red-contrib-modbus

* 1. Packages worth investigating
     1. node-red-node-sqlite

sudo npm install node-red-node-sqlite

* + 1. node-red-contrib-iiot-opcua

sudo npm install --unsafe-perm node-red-contrib-iiot-opcua

* + 1. sqlite3

sudo npm install sqlite3

* + 1. xlsx

sudo npm install xlsx

* + 1. pdf2json

sudo npm install pdf2json

* + 1. bluebird

sudo npm install bluebird

* + 1. sync

sudo npm install sync

* + 1. synchronize

sudo npm install –unsafe-perm synchronize

* + 1. edge

sudo npm install --unsafe-perm edge

* 1. Notes on packages with other versions of Node.JS
     1. The following packages didn’t work with Legacy Node.JS (4.x)
        1. node-red-contrib-iiot-opcua

sudo npm install node-red-contrib-iiot-opcua

* + 1. The following packages didn’t work with Node.JS 8.x (updated 1/20/2018)
       1. **node-red-contrib-function-npm** 🡪 This seems to install, but all package installations after installing this break because they’re looking for ‘ansistyles’ and can’t find it.

sudo npm install --unsafe-perm node-red-contrib-function-npm

* + - 1. **hummus** 🡪 This doesn’t seem to download

sudo npm install hummus

1. Add all non-Node-RED-specific packages to the settings.js file
   1. Open a terminal session and edit the settings.js with superuser creds

sudo gedit /home/nodered-daemon/.node-red/settings.js

* 1. Find the section that looks like this:

functionGlobalContext: {

// os:require('os'),

// octalbonescript:require('octalbonescript'),

// jfive:require("johnny-five"),

// j5board:require("johnny-five").Board({repl:false})

},

* 1. Replace it with this:

functionGlobalContext: {

os:require('os'),

fs:require('fs-extra'),

jszip:require('jszip'),

lodash:require('lodash'),

uuidv1:require('uuid/v1'),

uuidv3:require('uuid/v3'),

uuidv4:require('uuid/v4'),

uuidv5:require('uuid/v5'),

xmldom:require('xmldom'),

docxtemplater:require('docxtemplater'),

deasync:require('deasync'),

async:require('async'),

opcua:require('node-opcua'),

honcho:require('honcho')

},

* + 1. The global context variable names ideally should not contain special characters (i.e. hyphens)
    2. The last package line should NOT end in a comma. All other package lines should end in a comma.

1. Verify the base Node-RED install works as a service
   1. Reboot or restart the node-red service

sudo systemctl restart node-red

* 1. Open a browser and navigate to: http://localhost:1880
  2. Verify all desired packages have been loaded.

## Maintenance Commands

1. Start, Stop, or Restart the Service

sudo systemctl start node-red

sudo systemctl stop node-red

sudo systemctl restart node-red

## Miscellaneous Notes

1. Manual access to the Node-RED configuration files.
   1. To open the folder with the config files:
      1. Open a terminal session and run following command:

sudo nautilus /home/nodered-daemon/.node-red/

* + 1. Note ‘/home/nodered-daemon/’ is the WorkingDirectory from the service script above.
  1. Edit ‘settings.js’ to add addition require() statements to the global variables.

sudo gedit /home/nodered-daemon/.node-red/settings.js

* 1. Edit ‘flows\_$*MACHINENAME*.json’ to inject code into a Node-RED flow or the whole flow.
     1. Note: $*MACHINENAME* = environment variable containing the machine name.

sudo gedit /home/nodered-daemon/.node-red/flows\_$MACHINENAME.json