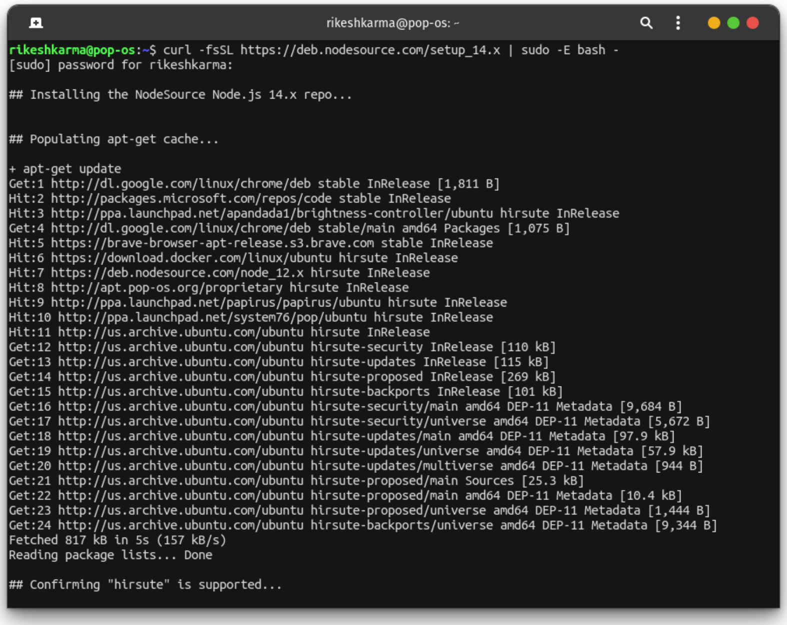
Node JS:

Node.js is an open-source, back-end JavaScript runtime environment that enables developers to execute JavaScript code without a browser. Built on Google Chrome's V8 JavaScript engine, Node.js is a popular platform for creating solutions that require speed and scalability.

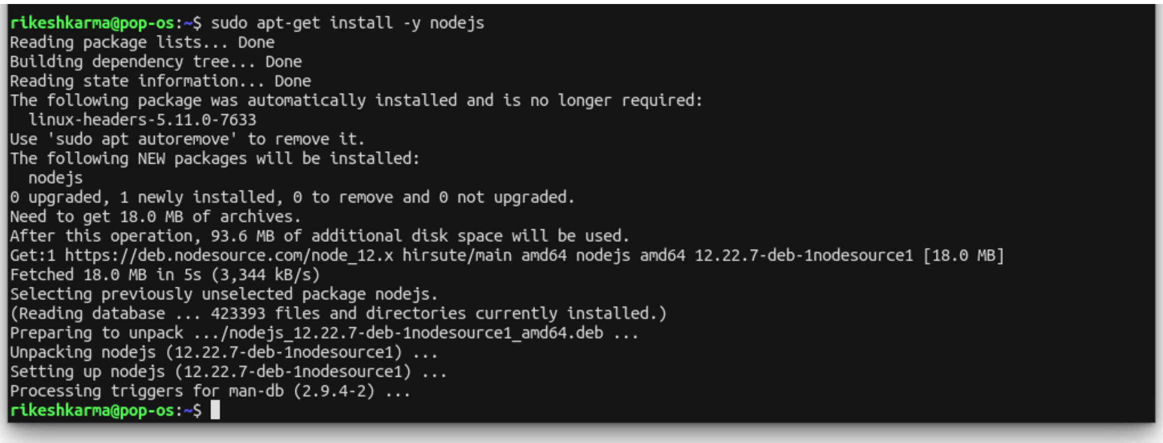
One of the core features of this runtime environment is that it is cross-platform. Developers can deploy it on Windows, macOS, and Linux. Node.js also supports both 32-bit and 64-bit environments. In addition to supporting the x86 CPU architecture, it can also run on specific ARM platforms. Node.js is also available as a Docker image and can also run on the IBM PowerLinux, System z, and AIX platforms.

- Install Node js on local VM.
    - Node.js gives us the option to install it using the package manager of our relevant Linux distribution. Due to its popularity, there is a Node.js install package for every popular Linux platform.
    - As my Linux Distribution is Ubuntu, I need to access the binaries via NodeSource by specifying the version number we need. The installation requires two commands as below:
        i. The first command that downloads the setup file via curl
            - curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -



```
rikeshkarma@pop-os:~$ curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
[sudo] password for rikeshkarma:

## Installing the NodeSource Node.js 14.x repo...


## Populating apt-get cache...

+ apt-get update
Get:1 http://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Hit:2 http://packages.microsoft.com/repos/code stable InRelease
Hit:3 http://ppa.launchpad.net/apandada1/brightness-controller/ubuntu hirsute InRelease
Get:4 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,075 B]
Hit:5 https://brave-browser-apt-release.s3.brave.com stable InRelease
Hit:6 https://download.docker.com/linux/ubuntu hirsute InRelease
Hit:7 https://deb.nodesource.com/node_12.x hirsute InRelease
Hit:8 http://apt.pop-os.org/proprietary hirsute InRelease
Hit:9 http://ppa.launchpad.net/papirus/papirus/ubuntu hirsute InRelease
Hit:10 http://ppa.launchpad.net/system76/pop/ubuntu hirsute InRelease
Hit:11 http://us.archive.ubuntu.com/ubuntu hirsute InRelease
Get:12 http://us.archive.ubuntu.com/ubuntu hirsute-security InRelease [110 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu hirsute-updates InRelease [115 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu hirsute-proposed InRelease [269 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu hirsute-backports InRelease [101 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu hirsute-security/main amd64 DEP-11 Metadata [9,684 B]
Get:17 http://us.archive.ubuntu.com/ubuntu hirsute-security/universe amd64 DEP-11 Metadata [5,672 B]
Get:18 http://us.archive.ubuntu.com/ubuntu hirsute-updates/main amd64 DEP-11 Metadata [97.9 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu hirsute-updates/universe amd64 DEP-11 Metadata [57.9 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu hirsute-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:21 http://us.archive.ubuntu.com/ubuntu hirsute-proposed/main Sources [25.3 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu hirsute-proposed/main amd64 DEP-11 Metadata [10.4 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu hirsute-proposed/universe amd64 DEP-11 Metadata [1,444 B]
Get:24 http://us.archive.ubuntu.com/ubuntu hirsute-backports/universe amd64 DEP-11 Metadata [9,344 B]
Fetched 817 kB in 5s (157 kB/s)
Reading package lists... Done

## Confirming "hirsute" is supported...
```

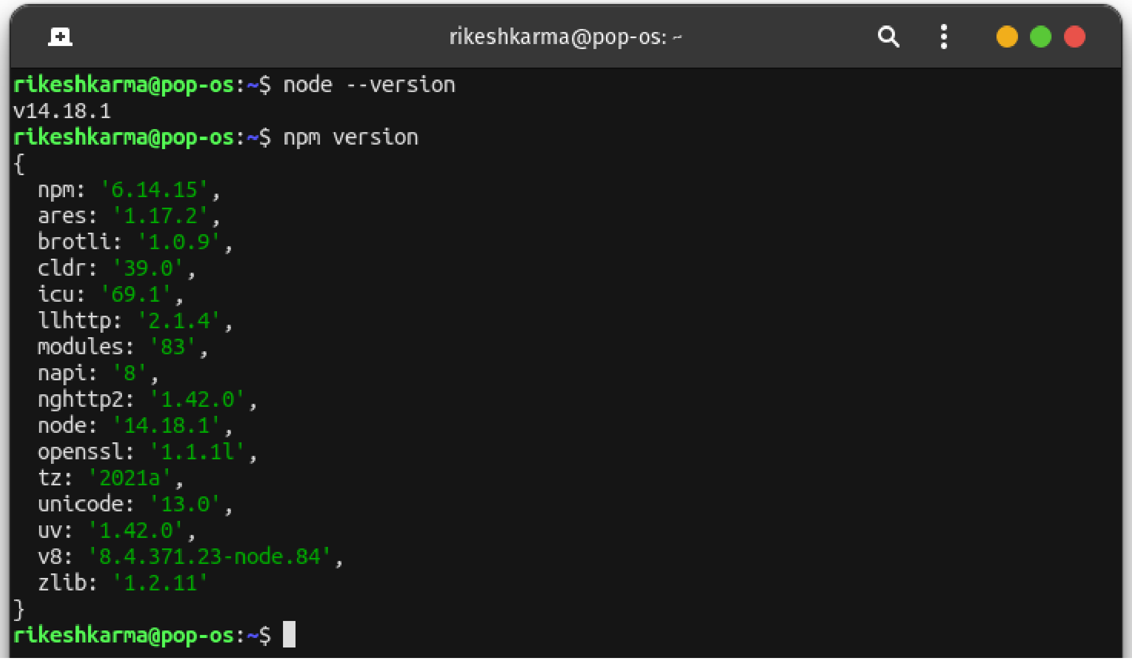
ii. And the second command installs Node.js on the system
    - sudo apt-get install -y nodejs



```
rikeshkarma@pop-os:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-headers-5.11.0-7633
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.0 MB of archives.
After this operation, 93.6 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_12.x hirsute/main amd64 nodejs amd64 12.22.7-deb-1nodesource1 [18.0 MB]
Fetched 18.0 MB in 5s (3,344 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 423393 files and directories currently installed.)
Preparing to unpack .../nodejs_12.22.7-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (12.22.7-deb-1nodesource1) ...
Setting up nodejs (12.22.7-deb-1nodesource1) ...
Processing triggers for man-db (2.9.4-2) ...
rikeshkarma@pop-os:~$
```

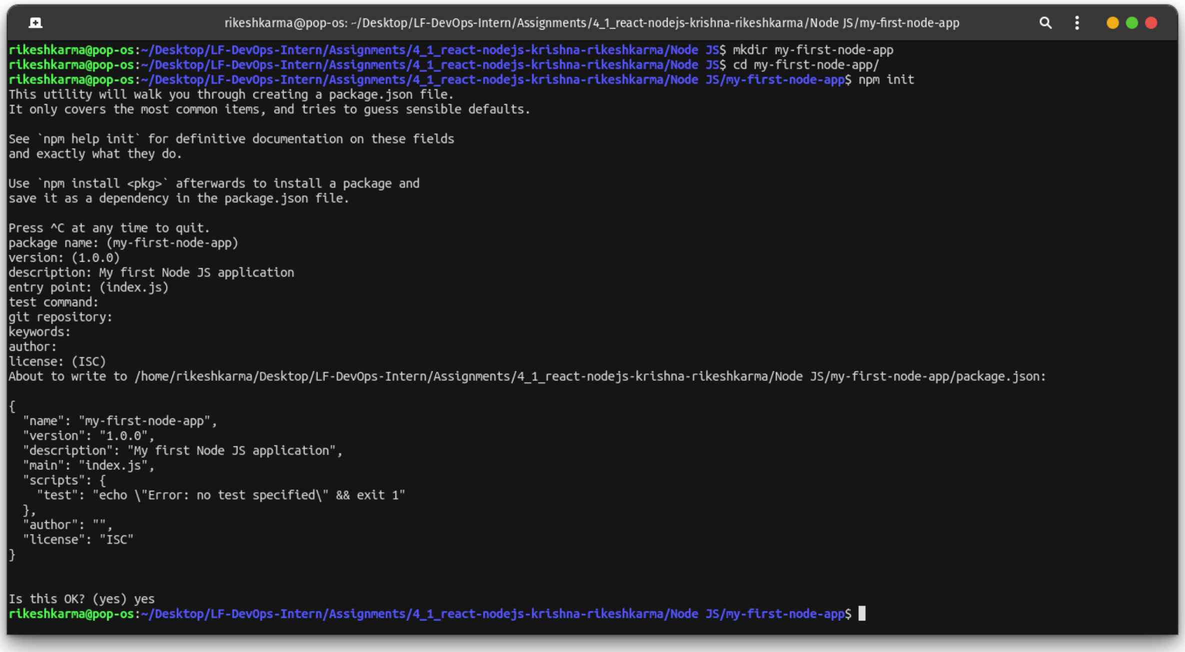
iii. Also, we can install the development tools such as **gcc-c++** and **make** that you need to have on your system to build native addons from npm.
    - *sudo apt-get install -y build-essential*

- ○ Now to verify our installation we use the following commands:
    i. node --version
    ii. npm --version



```
rikeshkarma@pop-os:~$ node --version
v14.18.1
rikeshkarma@pop-os:~$ npm version
{
  npm: '6.14.15',
  ares: '1.17.2',
  brotli: '1.0.9',
  cldr: '39.0',
  icu: '69.1',
  llhttp: '2.1.4',
  modules: '83',
  napi: '8',
  nghttp2: '1.42.0',
  node: '14.18.1',
  openssl: '1.1.1l',
  tz: '2021a',
  unicode: '13.0',
  uv: '1.42.0',
  v8: '8.4.371.23-node.84',
  zlib: '1.2.11'
}
rikeshkarma@pop-os:~$
```

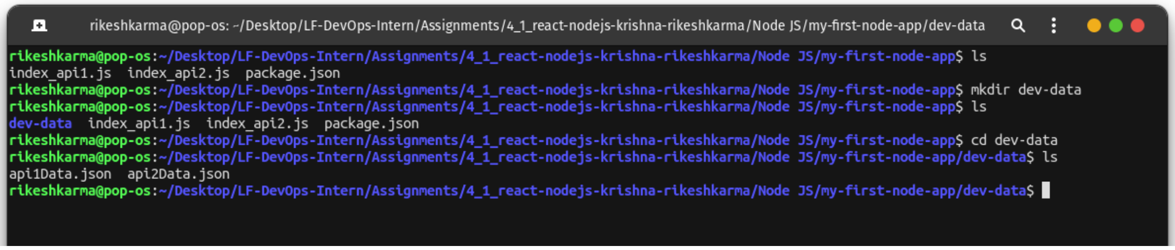
- Create 2 APIs running on ports 6080 and 7080 with messages "Hello Node JS" and "Node JS installed successfully" respectively.
    - First, we need to create a directory and navigate inside the directory and invoke the *npm init* command in order to initialize a node.js project. For this we run the following commands serially:
        i. mkdir my-first-node-app
        ii. cd my-first-node-app
        iii. npm init



rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS$ mkdir my-first-node-app
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS$ cd my-first-node-app/
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (my-first-node-app)
version: (1.0.0)
description: My first Node JS application
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/package.json:

{
  "name": "my-first-node-app",
  "version": "1.0.0",
  "description": "My first Node JS application",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ 
```

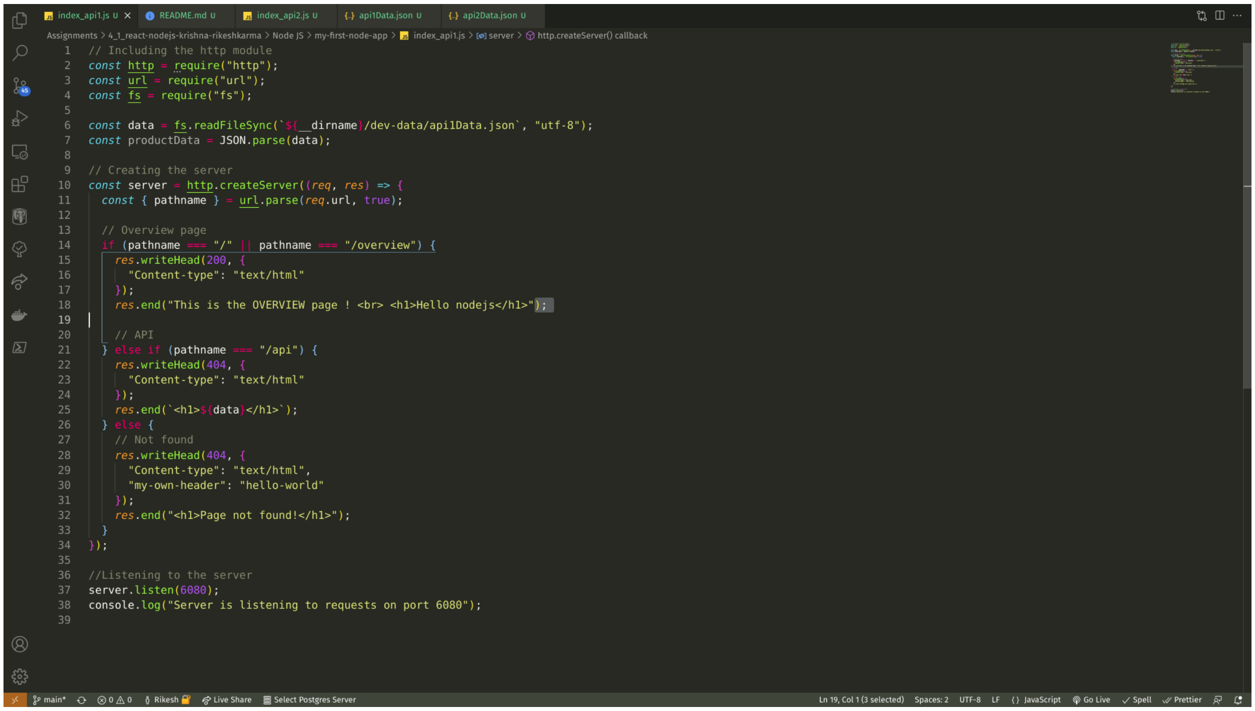
    - After we fill in some information while invoking the *npm init* command, we have initialized our new node js application. We have named out app “my-first-node-app”
    - Now we create two apps named ‘index_api1.js’ and ’index_api2.js’ as demanded by the question. And we create a folder named ‘dev-data’ where we store our APIs for both the index files named ‘api1Data.json’ and ’api2Data.json’.



rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/dev-data

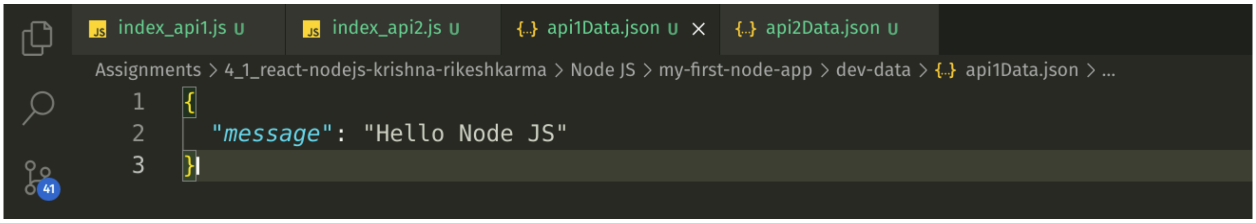
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ ls
index_api1.js  index_api2.js  package.json
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ mkdir dev-data
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ ls
dev-data  index_api1.js  index_api2.js  package.json
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ cd dev-data
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/dev-data$ ls
api1Data.json  api2Data.json
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/dev-data$ 
```

- We need to run our first API on port 6080 and our second API on port 7080 with the print messages "Hello Node JS" and "Node JS installed successfully" respectively.
- To access the API we need to type “/api” after the port number.
  i. For 1st API
     - Our *index_api1.js* file has been coded as:


index_api1.js U
README.md U
index_api2.js U
api1Data.json U
api2Data.json U
Assignments > 4_1_react-nodejs-krishna-rikeshkarma > Node JS > my-first-node-app > index_api1.js > server > http.createServer() callback
main* 0 0 Rikesh Live Share Select Postgres Server
Ln 19, Col 1 (3 selected) Spaces: 2 UTF-8 LF JavaScript Go Live Spell Prettier

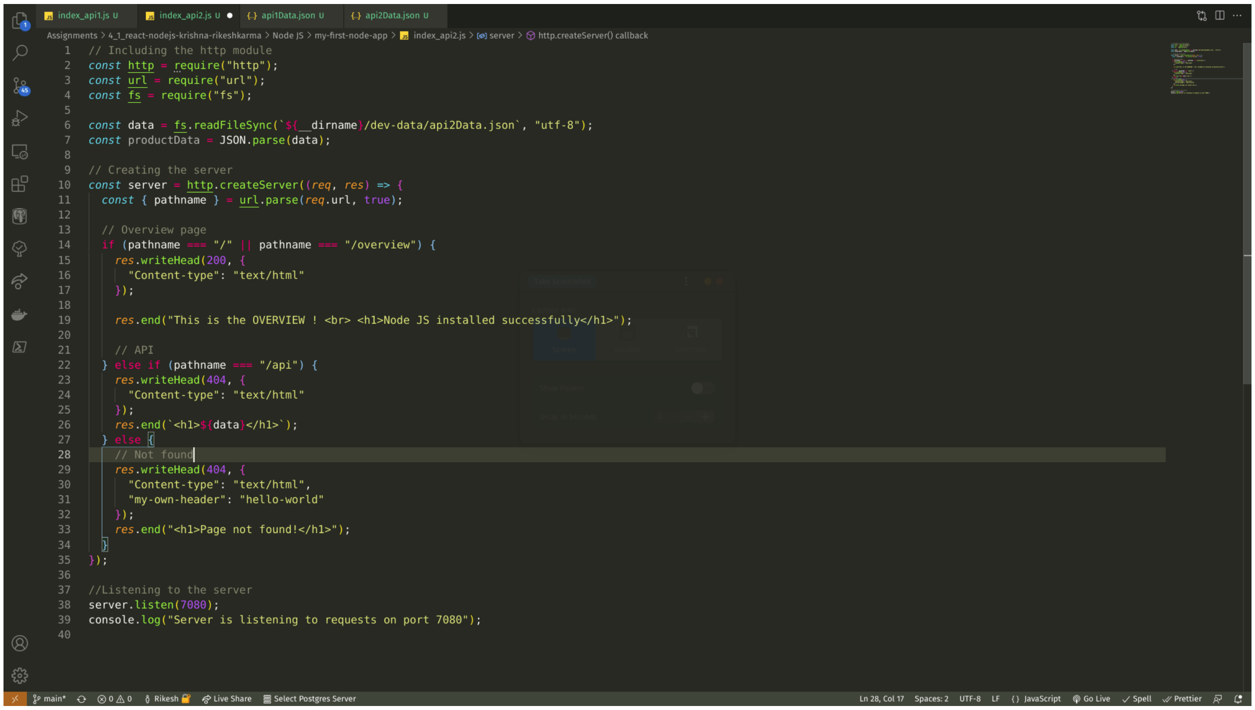
```javascript
// Including the http module
const http = require("http");
const url = require("url");
const fs = require("fs");

const data = fs.readFileSync(`${__dirname}/dev-data/api1Data.json`, "utf-8");
const productData = JSON.parse(data);

// Creating the server
const server = http.createServer((req, res) => {
  const { pathname } = url.parse(req.url, true);

  // Overview page
  if (pathname === "/" || pathname === "/overview") {
    res.writeHead(200, {
      "Content-type": "text/html"
    });
    res.end("This is the OVERVIEW page ! <br> <h1>Hello nodejs</h1>");

    // API
  } else if (pathname === "/api") {
    res.writeHead(404, {
      "Content-type": "text/html"
    });
    res.end(`<h1>${data}</h1>`);
  } else {
    // Not found
    res.writeHead(404, {
      "Content-type": "text/html",
      "my-own-header": "hello-world"
    });
    res.end("<h1>Page not found!</h1>");
  }
});

//Listening to the server
server.listen(6080);
console.log("Server is listening to requests on port 6080");
```

     - And the API as:


index_api1.js U
index_api2.js U
api1Data.json U
api2Data.json U
Assignments > 4_1_react-nodejs-krishna-rikeshkarma > Node JS > my-first-node-app > dev-data > api1Data.json > ...

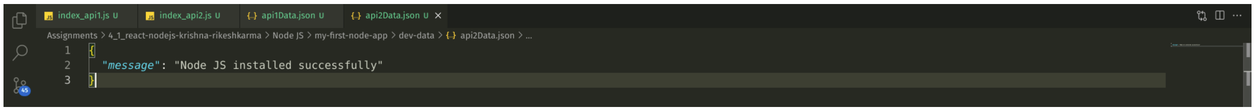
```json
{
  "message": "Hello Node JS"
}
```

  ii. For 2nd API
     - Our *index_api2.js* file has been coded as:


index_api1.js U
index_api2.js U
api1Data.json U
api2Data.json U
Assignments > 4_1_react-nodejs-krishna-rikeshkarma > Node JS > my-first-node-app > index_api2.js > server > http.createServer() callback
main* ⊗ 0 ⚠ 0 Rikesh Live Share Select Postgres Server
Ln 28, Col 17 Spaces: 2 UTF-8 LF {} JavaScript Go Live ✓ Spell ✓ Prettier

```javascript
// Including the http module
const http = require("http");
const url = require("url");
const fs = require("fs");

const data = fs.readFileSync(`${__dirname}/dev-data/api2Data.json`, "utf-8");
const productData = JSON.parse(data);

// Creating the server
const server = http.createServer((req, res) => {
  const { pathname } = url.parse(req.url, true);

  // Overview page
  if (pathname === "/" || pathname === "/overview") {
    res.writeHead(200, {
      "Content-type": "text/html"
    });

    res.end("This is the OVERVIEW ! <br> <h1>Node JS installed successfully</h1>");

    // API
  } else if (pathname === "/api") {
    res.writeHead(404, {
      "Content-type": "text/html"
    });
    res.end(`<h1>${data}</h1>`);
  } else {
    // Not found
    res.writeHead(404, {
      "Content-type": "text/html",
      "my-own-header": "hello-world"
    });
    res.end("<h1>Page not found!</h1>");
  }
});

//Listening to the server
server.listen(7080);
console.log("Server is listening to requests on port 7080");
```

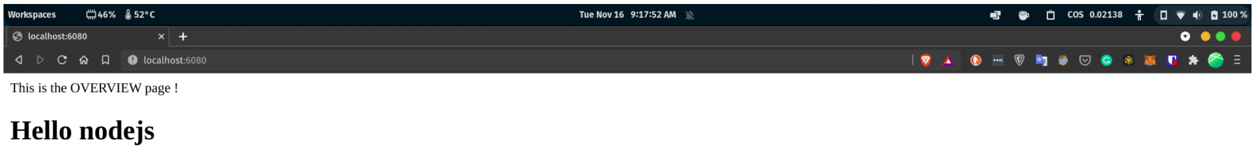
- And the API as:


index_api1.js U
index_api2.js U
api1Data.json U
api2Data.json U
Assignments > 4_1_react-nodejs-krishna-rikeshkarma > Node JS > my-first-node-app > dev-data > api2Data.json > ...

```json
{
  "message": "Node JS installed successfully"
}
```

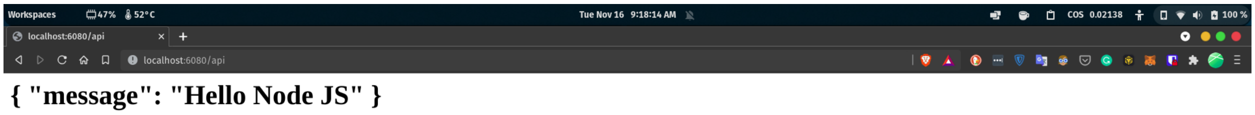
After all this, we can check if both the API are running on their own ports and printing the required messages.

➔ First API



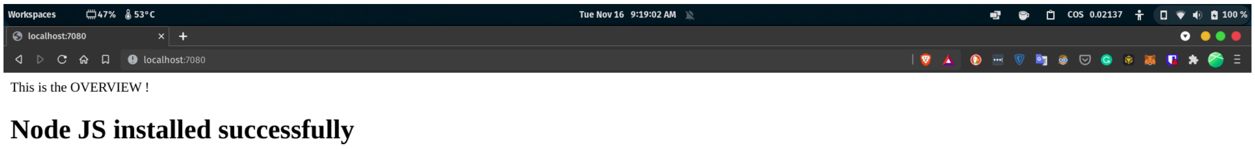
This is the OVERVIEW page !

Hello nodejs



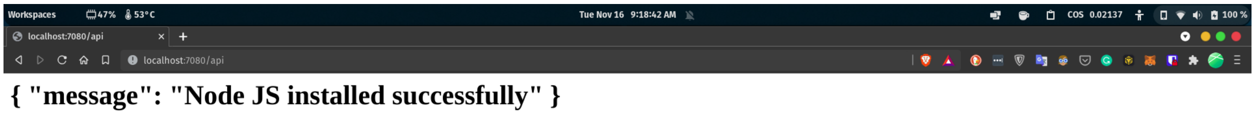
{ "message": "Hello Node JS" }

➔ Second API



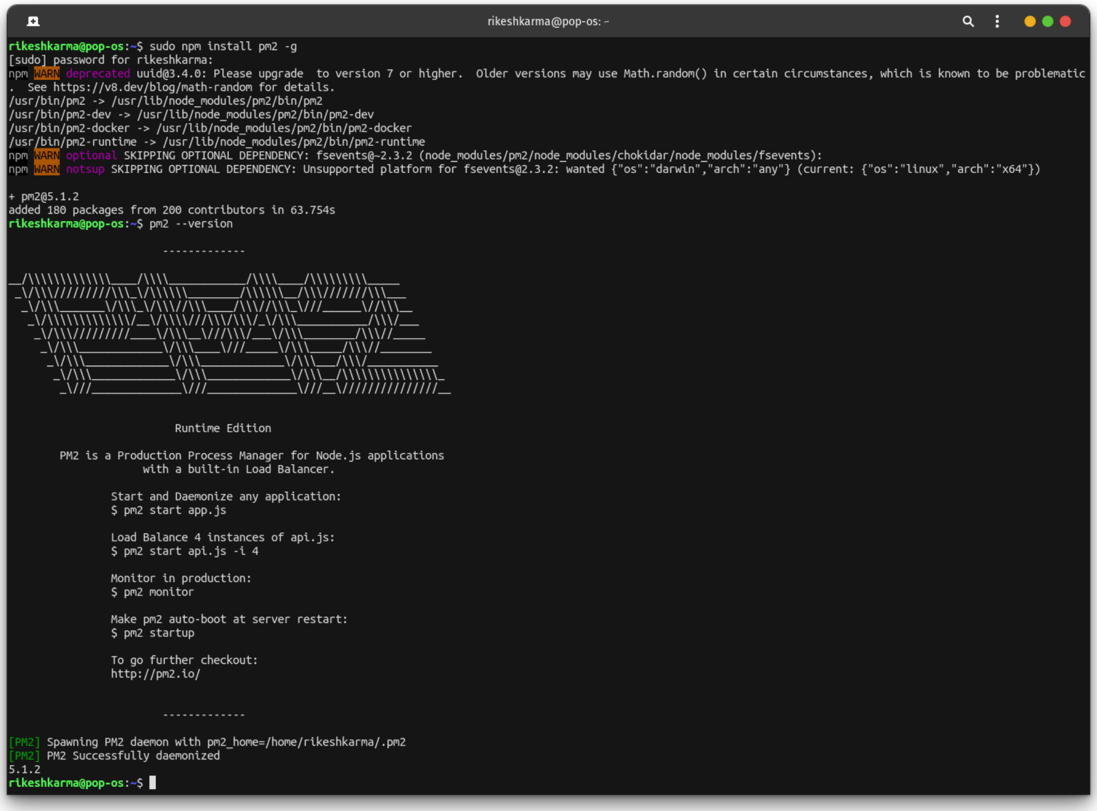
This is the OVERVIEW !

Node JS installed successfully



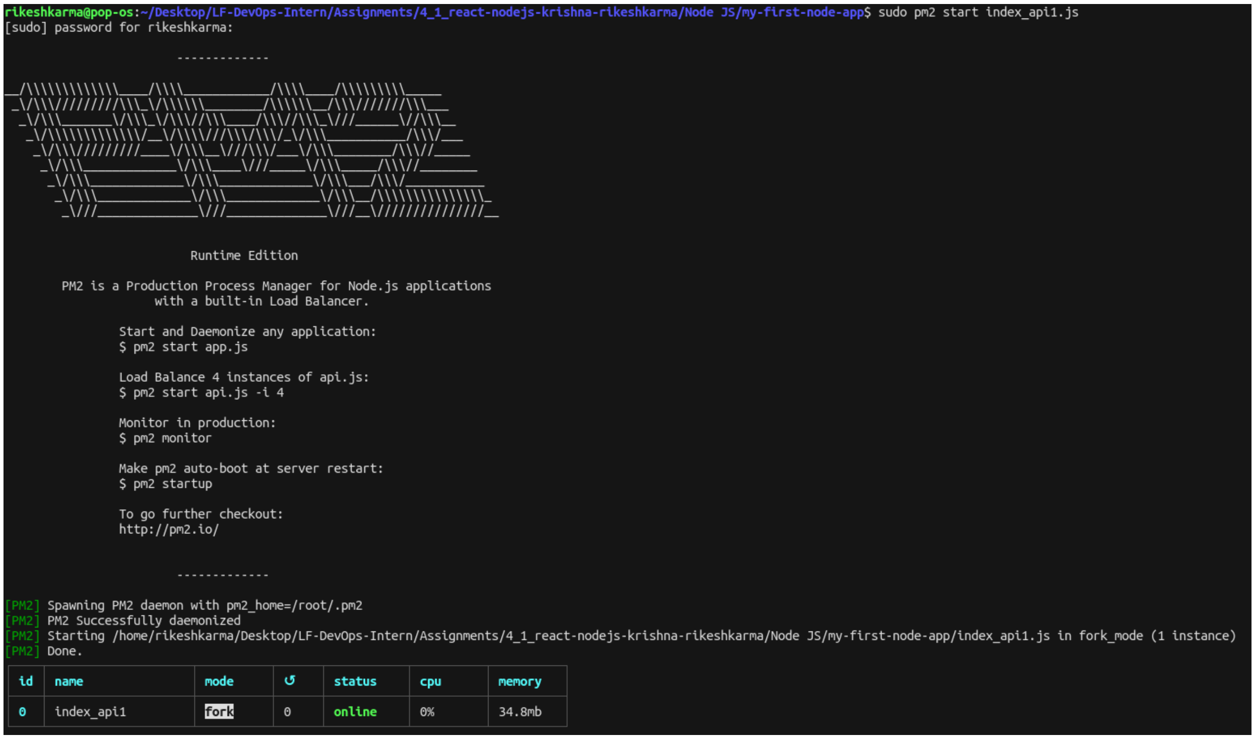
{ "message": "Node JS installed successfully" }

- Install the pm2 tool and create 4 clusters of both Nodes’.
  - To install the pm2 tool we invoke the following command in the terminal:
    1. *npm install pm2 -g*
    2. *We can use pm2 --version to verify the installation and version*



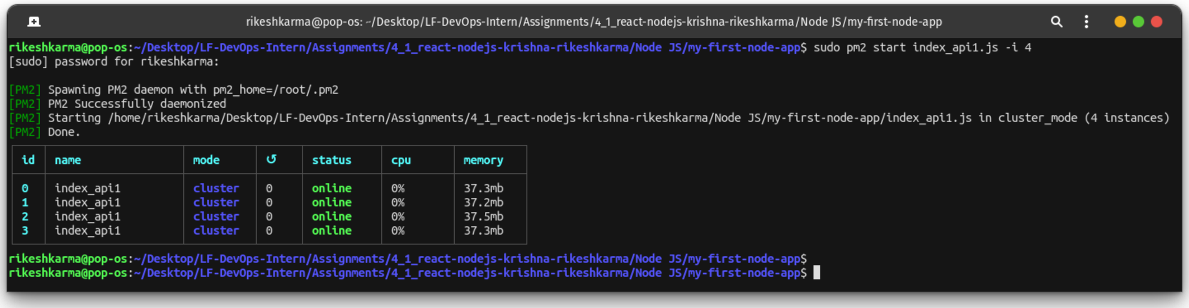
```
rikeshkarma@pop-os:~$ sudo npm install pm2 -g
[sudo] password for rikeshkarma:
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, which is known to be problematic
.  See https://v8.dev/blog/math-random for details.
/usr/bin/pm2 -> /usr/lib/node_modules/pm2/bin/pm2
/usr/bin/pm2-dev -> /usr/lib/node_modules/pm2/bin/pm2-dev
/usr/bin/pm2-docker -> /usr/lib/node_modules/pm2/bin/pm2-docker
/usr/bin/pm2-runtime -> /usr/lib/node_modules/pm2/bin/pm2-runtime
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules/pm2/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

+ pm2@5.1.2
added 180 packages from 200 contributors in 63.754s
rikeshkarma@pop-os:~$ pm2 --version

                        -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                          Runtime Edition

        PM2 is a Production Process Manager for Node.js applications
                     with a built-in Load Balancer.

                Start and Daemonize any application:
                $ pm2 start app.js

                Load Balance 4 instances of api.js:
                $ pm2 start api.js -i 4

                Monitor in production:
                $ pm2 monitor

                Make pm2 auto-boot at server restart:
                $ pm2 startup

                To go further checkout:
                http://pm2.io/


                        -------------

[PM2] Spawning PM2 daemon with pm2_home=/home/rikeshkarma/.pm2
[PM2] PM2 Successfully daemonized
5.1.2
rikeshkarma@pop-os:~$
```

- Create 4 clusters on the first node/ API.
  1. Let's first start the pm2 tool using the following command:
     - sudo pm2 start index_api1.js



```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 start index_api1.js
[sudo] password for rikeshkarma:

                        -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                          Runtime Edition

        PM2 is a Production Process Manager for Node.js applications
                     with a built-in Load Balancer.

                Start and Daemonize any application:
                $ pm2 start app.js

                Load Balance 4 instances of api.js:
                $ pm2 start api.js -i 4

                Monitor in production:
                $ pm2 monitor

                Make pm2 auto-boot at server restart:
                $ pm2 startup

                To go further checkout:
                http://pm2.io/


                        -------------

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/index_api1.js in fork_mode (1 instance)
[PM2] Done.
│ id │ name         │ mode     │ ↺    │ status    │ cpu      │ memory   │
│ 0  │ index_api1   │ fork     │ 0    │ online    │ 0%       │ 34.8mb   │
```

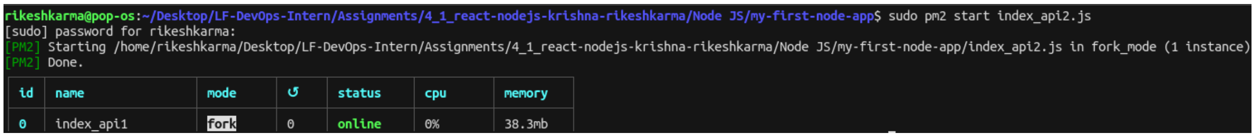
  2. To start the application in 4 cluster mode let's use the following command:
     - sudo pm2 start index_api1.js -i 4



rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 start index_api1.js -i 4
[sudo] password for rikeshkarma:

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/index_api1.js in cluster_mode (4 instances)
[PM2] Done.
│ id │ name         │ mode     │ ↺    │ status    │ cpu      │ memory   │
│ 0  │ index_api1   │ cluster  │ 0    │ online    │ 0%       │ 37.3mb   │
│ 1  │ index_api1   │ cluster  │ 0    │ online    │ 0%       │ 37.2mb   │
│ 2  │ index_api1   │ cluster  │ 0    │ online    │ 0%       │ 37.5mb   │
│ 3  │ index_api1   │ cluster  │ 0    │ online    │ 0%       │ 37.3mb   │
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$
```

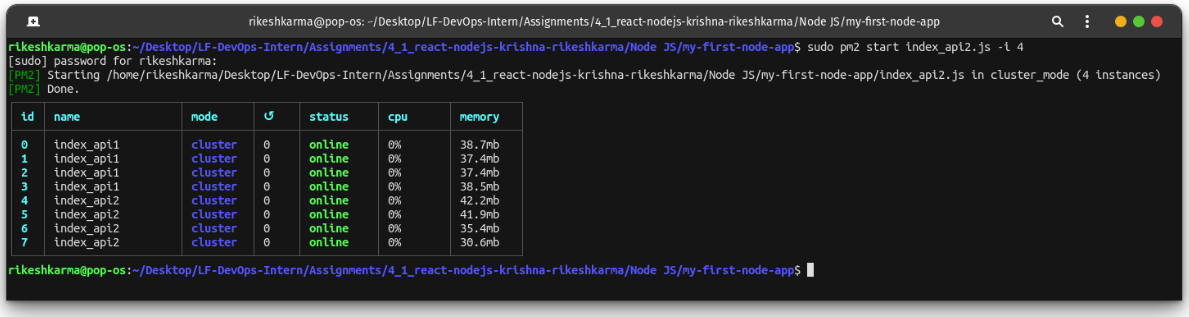
- Create 4 clusters on the second node/ API.
  1. Let's first start the pm2 tool using the following command:
     - sudo pm2 start index_api2.js



```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 start index_api2.js
[sudo] password for rikeshkarma:
[PM2] Starting /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/index_api2.js in fork_mode (1 instance)
[PM2] Done.
│ id │ name         │ mode     │ ↺    │ status    │ cpu      │ memory   │
│ 0  │ index_api1   │ fork     │ 0    │ online    │ 0%       │ 38.3mb   │
```

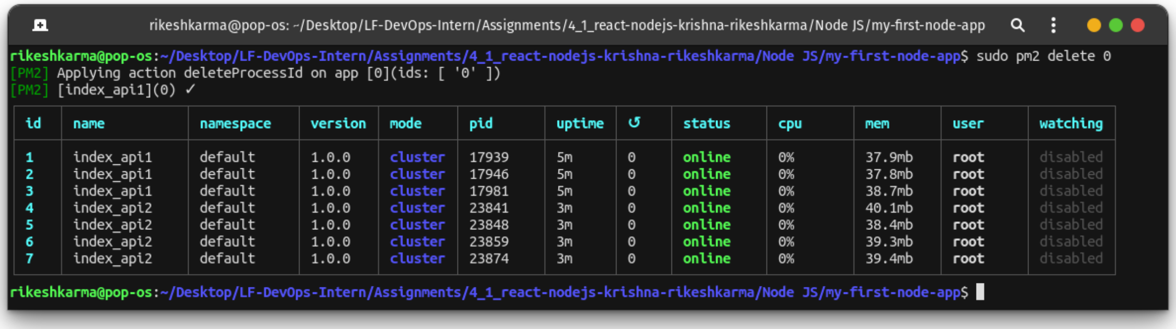
ii. To start the application in 4 cluster mode let's use the following command:

- sudo pm2 start index_api2.js -i 4



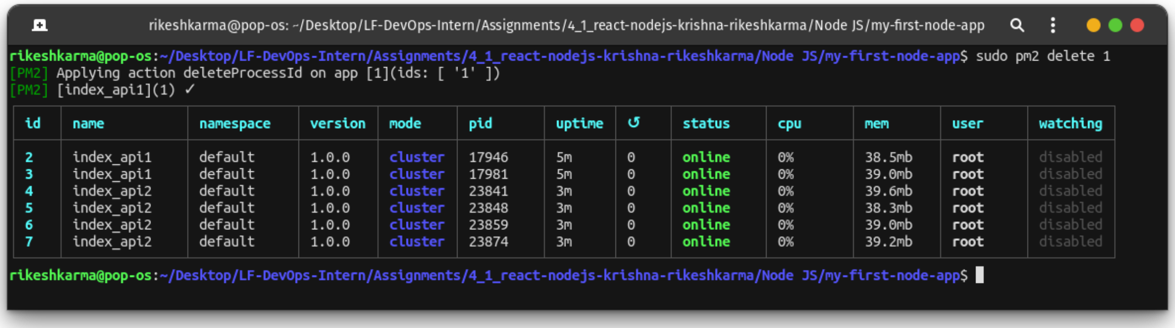
```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 start index_api2.js -i 4
[sudo] password for rikeshkarma:
[PM2] Starting /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app/index_api2.js in cluster_mode (4 instances)
[PM2] Done.
┌────┬───────────────┬─────────┬───┬─────────┬──────┬──────────┐
│ id │ name          │ mode    │ ↺ │ status  │ cpu  │ memory   │
├────┼───────────────┼─────────┼───┼─────────┼──────┼──────────┤
│ 0  │ index_api1    │ cluster │ 0 │ online  │ 0%   │ 38.7mb   │
│ 1  │ index_api1    │ cluster │ 0 │ online  │ 0%   │ 37.4mb   │
│ 2  │ index_api1    │ cluster │ 0 │ online  │ 0%   │ 37.4mb   │
│ 3  │ index_api1    │ cluster │ 0 │ online  │ 0%   │ 38.5mb   │
│ 4  │ index_api2    │ cluster │ 0 │ online  │ 0%   │ 42.2mb   │
│ 5  │ index_api2    │ cluster │ 0 │ online  │ 0%   │ 41.9mb   │
│ 6  │ index_api2    │ cluster │ 0 │ online  │ 0%   │ 35.4mb   │
│ 7  │ index_api2    │ cluster │ 0 │ online  │ 0%   │ 30.6mb   │
└────┴───────────────┴─────────┴───┴─────────┴──────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ █
```

- Delete all 4 clusters one-by-one
  - To delete all 4 clusters one by one of both the APIs/ Node we invoke the following commands serially:
    i. sudo pm2 delete 0



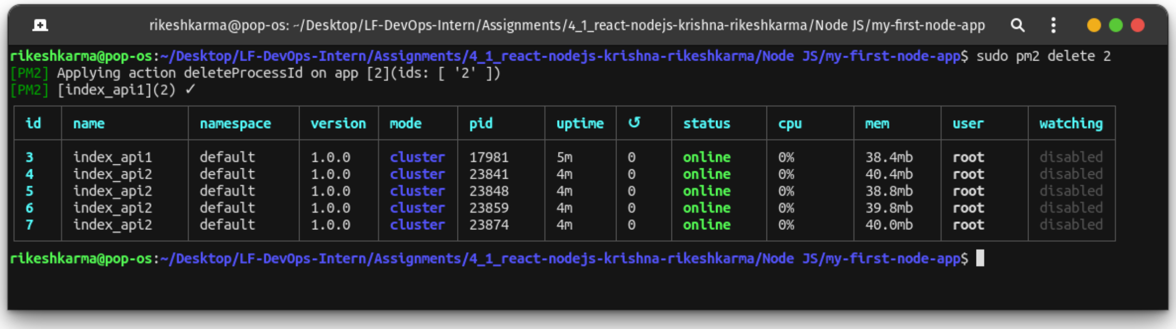
```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: [ '0' ])
[PM2] [index_api1](0) ✓
┌────┬───────────────┬───────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐
│ id │ name          │ namespace │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
├────┼───────────────┼───────────┼─────────┼─────────┼──────────┼────────┼──────┼───────────┼──────────┼──────────┼──────────┼──────────┤
│ 1  │ index_api1    │ default   │ 1.0.0   │ cluster │ 17939    │ 5m     │ 0    │ online    │ 0%       │ 37.9mb   │ root     │ disabled │
│ 2  │ index_api1    │ default   │ 1.0.0   │ cluster │ 17946    │ 5m     │ 0    │ online    │ 0%       │ 37.8mb   │ root     │ disabled │
│ 3  │ index_api1    │ default   │ 1.0.0   │ cluster │ 17981    │ 5m     │ 0    │ online    │ 0%       │ 38.7mb   │ root     │ disabled │
│ 4  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23841    │ 3m     │ 0    │ online    │ 0%       │ 40.1mb   │ root     │ disabled │
│ 5  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23848    │ 3m     │ 0    │ online    │ 0%       │ 38.4mb   │ root     │ disabled │
│ 6  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23859    │ 3m     │ 0    │ online    │ 0%       │ 39.3mb   │ root     │ disabled │
│ 7  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23874    │ 3m     │ 0    │ online    │ 0%       │ 39.4mb   │ root     │ disabled │
└────┴───────────────┴───────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ █
```

ii. sudo pm2 delete 1



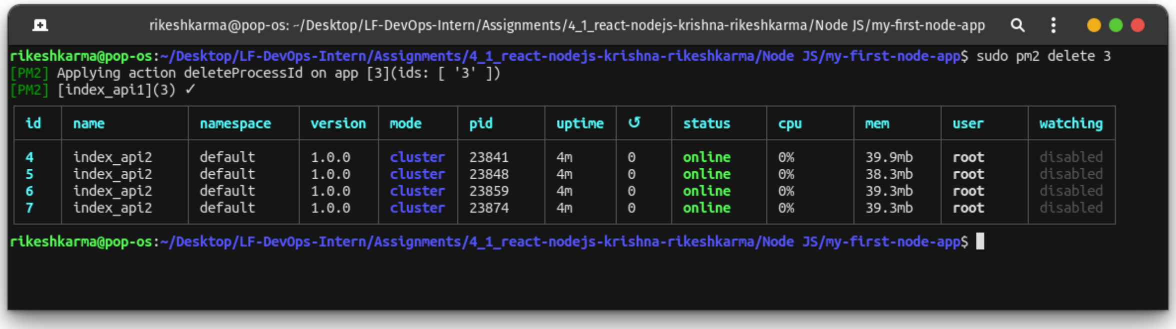
```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 1
[PM2] Applying action deleteProcessId on app [1](ids: [ '1' ])
[PM2] [index_api1](1) ✓
┌────┬───────────────┬───────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐
│ id │ name          │ namespace │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
├────┼───────────────┼───────────┼─────────┼─────────┼──────────┼────────┼──────┼───────────┼──────────┼──────────┼──────────┼──────────┤
│ 2  │ index_api1    │ default   │ 1.0.0   │ cluster │ 17946    │ 5m     │ 0    │ online    │ 0%       │ 38.5mb   │ root     │ disabled │
│ 3  │ index_api1    │ default   │ 1.0.0   │ cluster │ 17981    │ 5m     │ 0    │ online    │ 0%       │ 39.0mb   │ root     │ disabled │
│ 4  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23841    │ 3m     │ 0    │ online    │ 0%       │ 39.6mb   │ root     │ disabled │
│ 5  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23848    │ 3m     │ 0    │ online    │ 0%       │ 38.3mb   │ root     │ disabled │
│ 6  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23859    │ 3m     │ 0    │ online    │ 0%       │ 39.0mb   │ root     │ disabled │
│ 7  │ index_api2    │ default   │ 1.0.0   │ cluster │ 23874    │ 3m     │ 0    │ online    │ 0%       │ 39.2mb   │ root     │ disabled │
└────┴───────────────┴───────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ █
```

iii. sudo pm2 delete 2


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

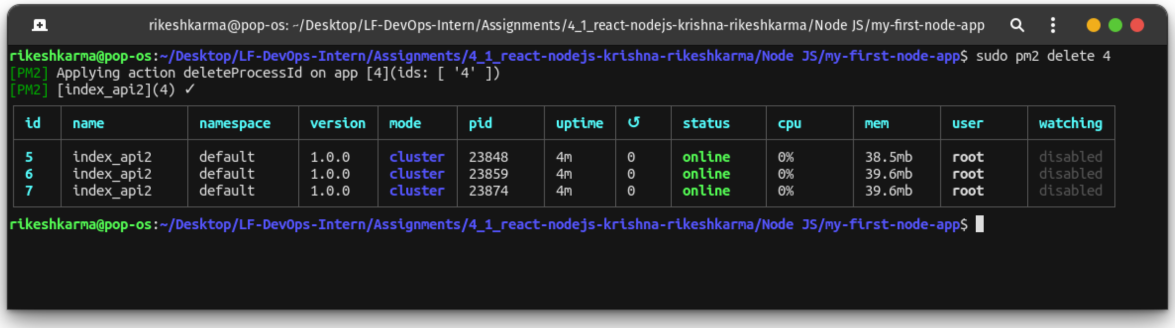
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 2
[PM2] Applying action deleteProcessId on app [2](ids: [ '2' ])
[PM2] [index_api1](2) ✓
│ id │ name       │ namespace │ version │ mode    │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 3  │ index_api1 │ default   │ 1.0.0   │ cluster │ 17981 │ 5m     │ 0 │ online │ 0%  │ 38.4mb │ root │ disabled │
│ 4  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23841 │ 4m     │ 0 │ online │ 0%  │ 40.4mb │ root │ disabled │
│ 5  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23848 │ 4m     │ 0 │ online │ 0%  │ 38.8mb │ root │ disabled │
│ 6  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23859 │ 4m     │ 0 │ online │ 0%  │ 39.8mb │ root │ disabled │
│ 7  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23874 │ 4m     │ 0 │ online │ 0%  │ 40.0mb │ root │ disabled │
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ 
```

iv. sudo pm2 delete 3


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

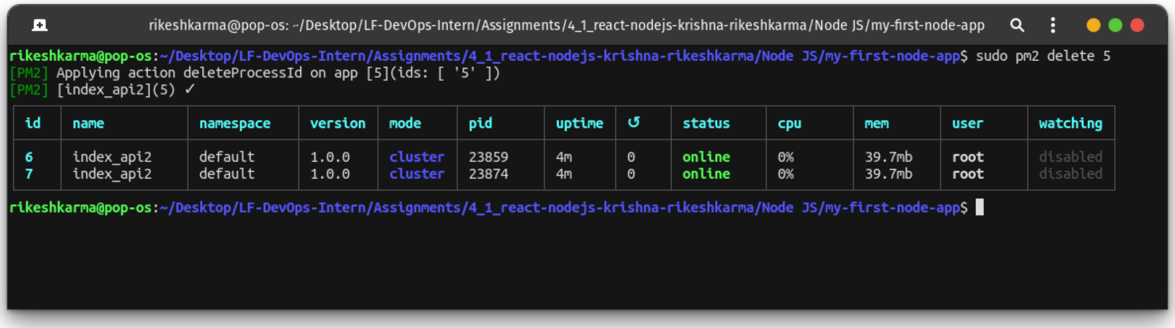
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 3
[PM2] Applying action deleteProcessId on app [3](ids: [ '3' ])
[PM2] [index_api1](3) ✓
│ id │ name       │ namespace │ version │ mode    │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 4  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23841 │ 4m     │ 0 │ online │ 0%  │ 39.9mb │ root │ disabled │
│ 5  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23848 │ 4m     │ 0 │ online │ 0%  │ 38.3mb │ root │ disabled │
│ 6  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23859 │ 4m     │ 0 │ online │ 0%  │ 39.3mb │ root │ disabled │
│ 7  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23874 │ 4m     │ 0 │ online │ 0%  │ 39.3mb │ root │ disabled │
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ 
```

v. sudo pm2 delete 4


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

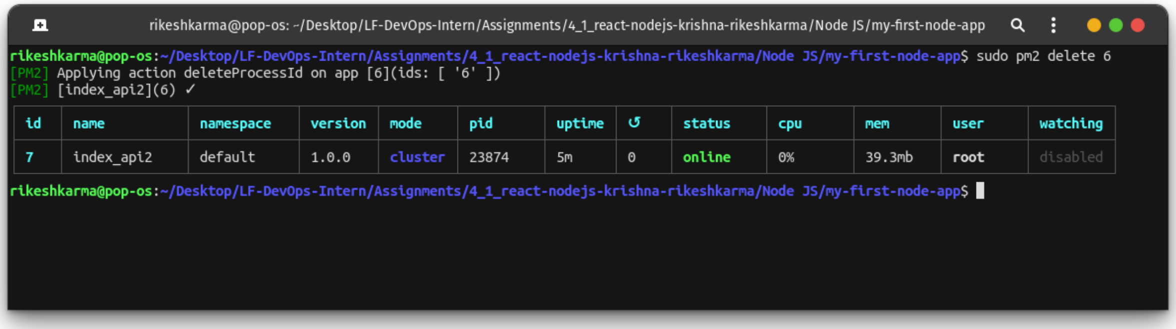
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 4
[PM2] Applying action deleteProcessId on app [4](ids: [ '4' ])
[PM2] [index_api2](4) ✓
│ id │ name       │ namespace │ version │ mode    │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 5  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23848 │ 4m     │ 0 │ online │ 0%  │ 38.5mb │ root │ disabled │
│ 6  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23859 │ 4m     │ 0 │ online │ 0%  │ 39.6mb │ root │ disabled │
│ 7  │ index_api2 │ default   │ 1.0.0   │ cluster │ 23874 │ 4m     │ 0 │ online │ 0%  │ 39.6mb │ root │ disabled │
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ 
```

vi. sudo pm2 delete 5


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

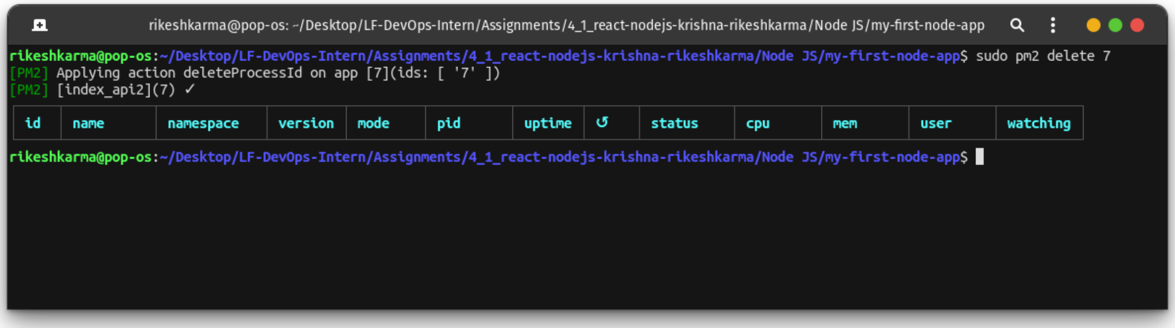
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 5
[PM2] Applying action deleteProcessId on app [5](ids: [ '5' ])
[PM2] [index_api2](5) ✓
┌────┬────────────────┬─────────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐
│ id │ name           │ namespace   │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
├────┼────────────────┼─────────────┼─────────┼─────────┼──────────┼────────┼──────┼───────────┼──────────┼──────────┼──────────┼──────────┤
│ 6  │ index_api2     │ default     │ 1.0.0   │ cluster │ 23859    │ 4m     │ 0    │ online    │ 0%       │ 39.7mb   │ root     │ disabled │
│ 7  │ index_api2     │ default     │ 1.0.0   │ cluster │ 23874    │ 4m     │ 0    │ online    │ 0%       │ 39.7mb   │ root     │ disabled │
└────┴────────────────┴─────────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ ▊
```

vii. sudo pm2 delete 6


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 6
[PM2] Applying action deleteProcessId on app [6](ids: [ '6' ])
[PM2] [index_api2](6) ✓
┌────┬────────────────┬─────────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐
│ id │ name           │ namespace   │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
├────┼────────────────┼─────────────┼─────────┼─────────┼──────────┼────────┼──────┼───────────┼──────────┼──────────┼──────────┼──────────┤
│ 7  │ index_api2     │ default     │ 1.0.0   │ cluster │ 23874    │ 5m     │ 0    │ online    │ 0%       │ 39.3mb   │ root     │ disabled │
└────┴────────────────┴─────────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ ▊
```

viii. sudo pm2 delete 7


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app

```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ sudo pm2 delete 7
[PM2] Applying action deleteProcessId on app [7](ids: [ '7' ])
[PM2] [index_api2](7) ✓
┌────┬───────────┬─────────────┬─────────┬─────────┬──────────┬────────┬──────┬───────────┬──────────┬──────────┬──────────┬──────────┐
│ id │ name      │ namespace   │ version │ mode    │ pid      │ uptime │ ↺    │ status    │ cpu      │ mem      │ user     │ watching │
└────┴───────────┴─────────────┴─────────┴─────────┴──────────┴────────┴──────┴───────────┴──────────┴──────────┴──────────┴──────────┘
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/Node JS/my-first-node-app$ ▊
```

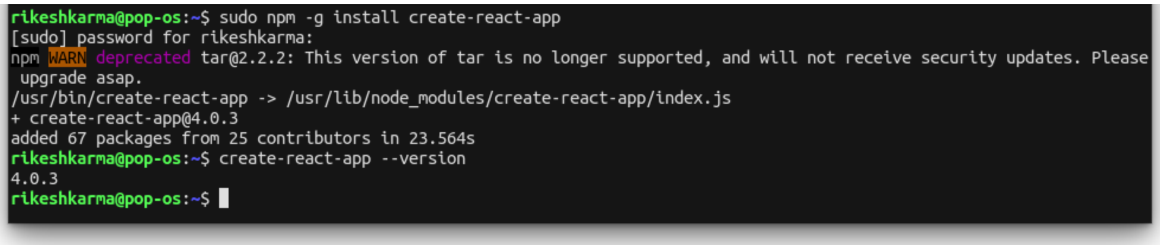
Now all the clusters of both the applications have been deleted one by one serially.

React JS:

Developed by Facebook in 2011, React (also referred to as ReactJS) is a Javascript library used for creating fast and interactive user interfaces. At this time, it’s the most popular Javascript library for developing user interfaces.

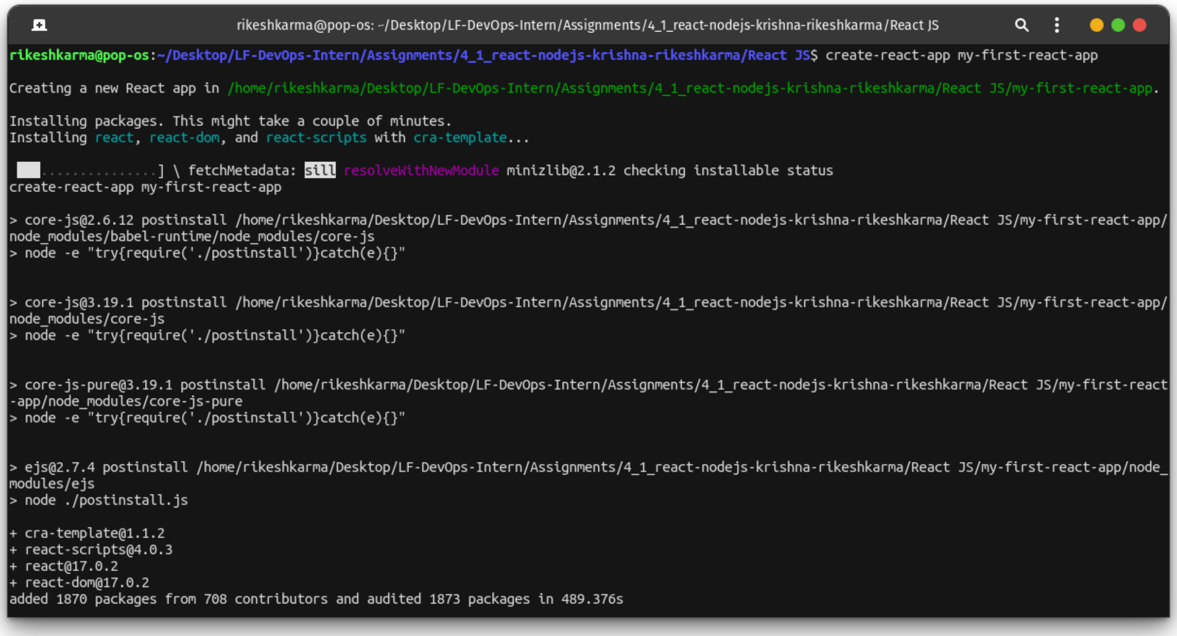
Its popularity stems from its flexibility and simplicity, making it the first choice in the development of mobile apps and web applications. More than 90,000 sites use React including tech giants such as Facebook, Netflix, Instagram, Airbnb, and Twitter to list a few

- Install React.js
    - Before we install **React JS**,
        - We need **npm** (node package manager), a command-line tool used for interacting with Javascript packages, that allows users to install, update, and manage Javascript tools and libraries. Also, npm is an online open-source software registry that hosts over 800,000 Node.JS packages.
        - Npm comes along with Node.JS so we don’t need to install it manually as we have already installed it in the previous task.
    - Installing create-react-app Utility
        - create-react-app is a utility that enables you to set up all the tools required to create a React Application. It saves you a great deal of time and energy setting everything from scratch and gives you the head start needed.
        - To install the tool, run the following npm command:
            - *sudo npm -g install create-react-app*



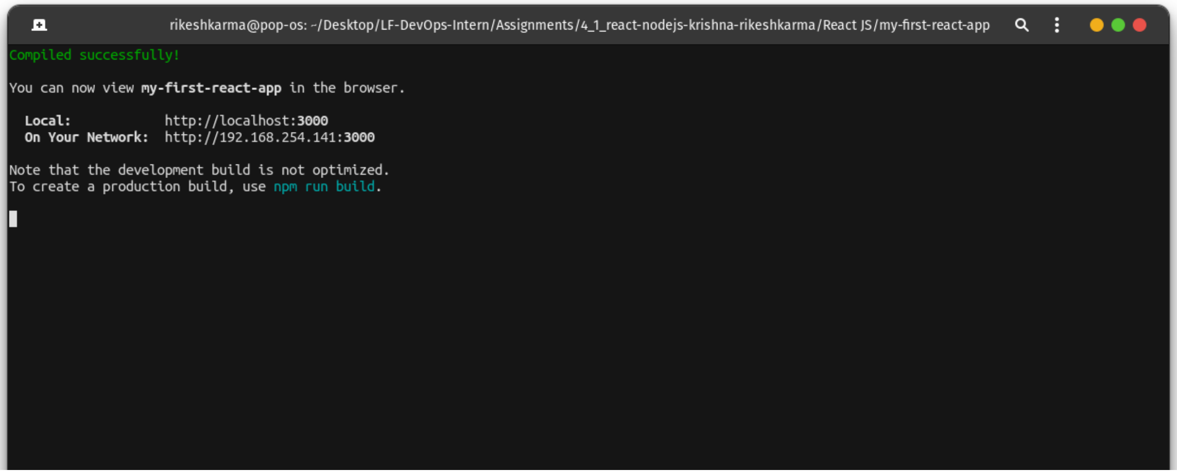
```
rikeshkarma@pop-os:~$ sudo npm -g install create-react-app
[sudo] password for rikeshkarma:
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please
 upgrade asap.
/usr/bin/create-react-app -> /usr/lib/node_modules/create-react-app/index.js
+ create-react-app@4.0.3
added 67 packages from 25 contributors in 23.564s
rikeshkarma@pop-os:~$ create-react-app --version
4.0.3
rikeshkarma@pop-os:~$ 
```

- Creating a React Application and print message "Hello React.js"
    - To create a react app we need to invoke the following command followed by the name of the app we want it to be:
        - *create-react-app my-first-react-app*


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS

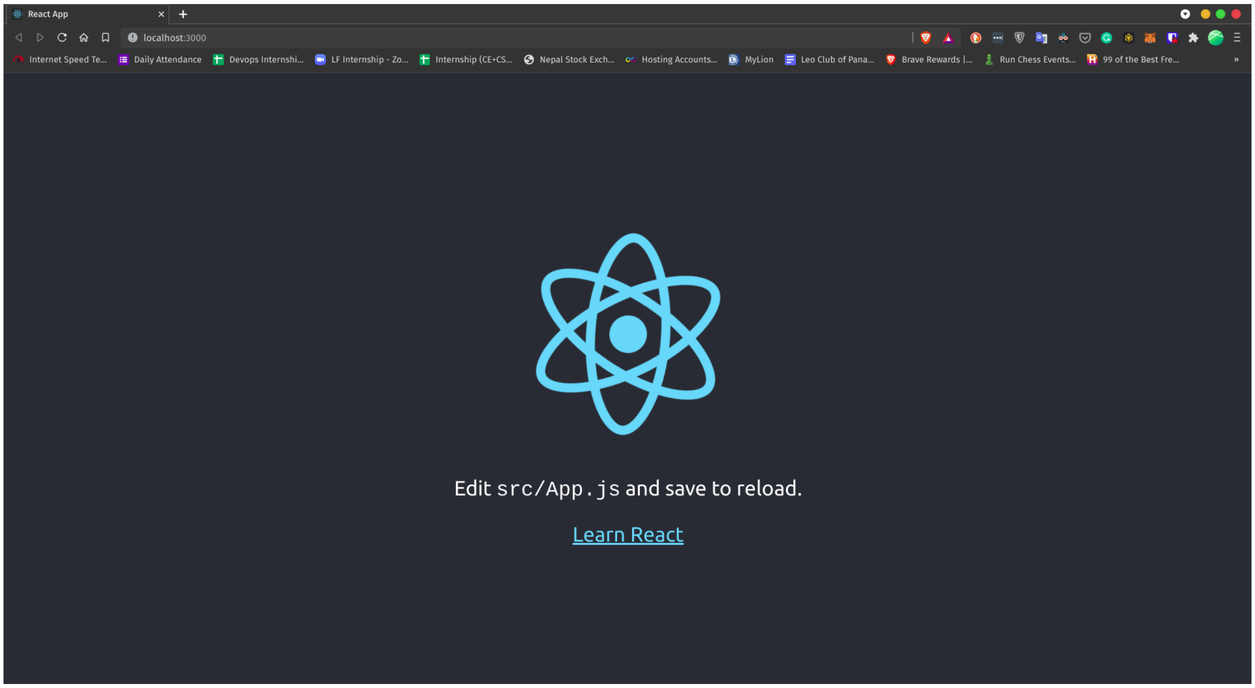
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS$ create-react-app my-first-react-app

Creating a new React app in /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

 ██...............] \ fetchMetadata: sill resolveWithNewModule minizlib@2.1.2 checking installable status
create-react-app my-first-react-app

> core-js@2.6.12 postinstall /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/
node_modules/babel-runtime/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"


> core-js@3.19.1 postinstall /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/
node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"


> core-js-pure@3.19.1 postinstall /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react
-app/node_modules/core-js-pure
> node -e "try{require('./postinstall')}catch(e){}"


> ejs@2.7.4 postinstall /home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/node_
modules/ejs
> node ./postinstall.js

+ cra-template@1.1.2
+ react-scripts@4.0.3
+ react@17.0.2
+ react-dom@17.0.2
added 1870 packages from 708 contributors and audited 1873 packages in 489.376s
```

- After we create a new react app, we need to navigate into the app directory in order to run the application. For this we use the following command:
  - *cd my-first-react-app*
  - *npm start*


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app

```
Compiled successfully!

You can now view my-first-react-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.254.141:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

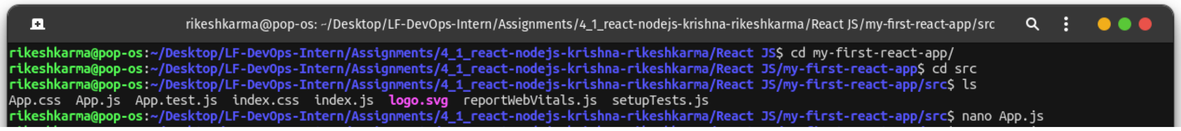
- Now we can go to our browser and check if our app is being served.



Edit `src/App.js` and save to reload.

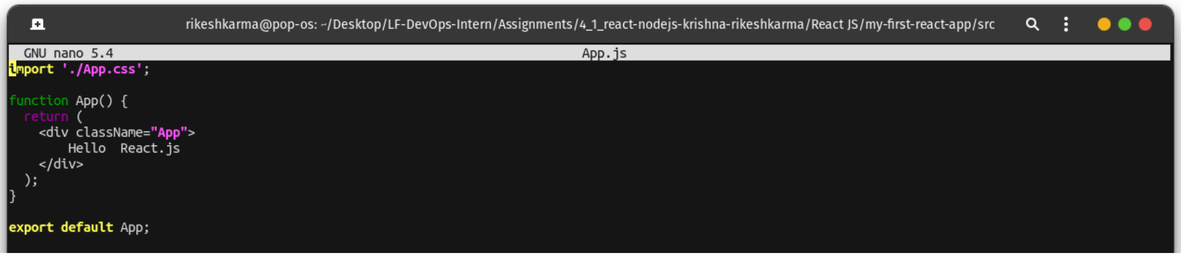
Learn React

    - This shows that the default React app is up and running. Now as per the question we need this react app to print the message “Hello React.js”.
        - We need to edit the *App.js* file located inside my-first-react-app/src


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/src

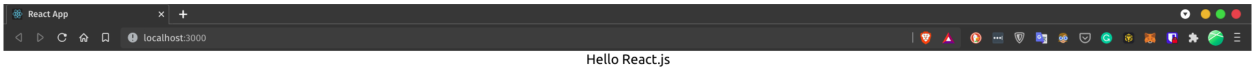
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS$ cd my-first-react-app/
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app$ cd src
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/src$ ls
App.css  App.js  App.test.js  index.css  index.js  logo.svg  reportWebVitals.js  setupTests.js
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/src$ nano App.js
```

        - Then configure the file as per our need, here need to print the message “Hello React.js”


rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app/src
GNU nano 5.4
App.js

```
import './App.css';

function App() {
  return (
    <div className="App">
        Hello  React.js
    </div>
  );
}

export default App;
```

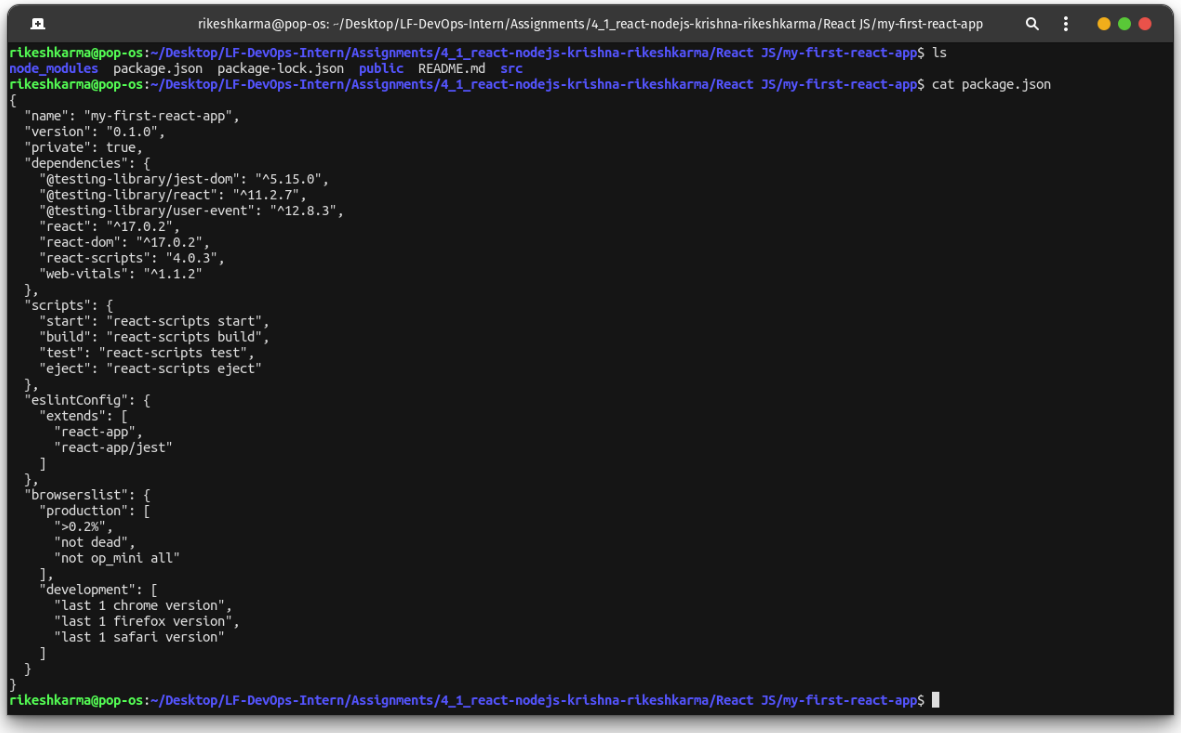
        - Now we can use *npm start, to check if the message is printing.*



Hello React.js

- Change the default port 3000 to 3001

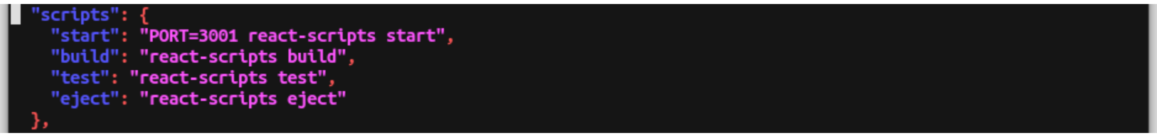
- Everything is well and running. We can see that the default port is 3000, so we need to change it to 3001.
  - For this, we need to change the scripts in the *package.json* file.



rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app

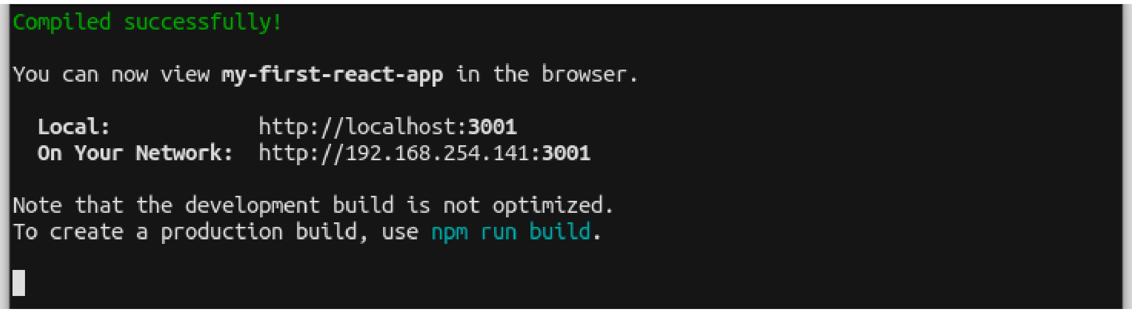
```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app$ ls
node_modules  package.json  package-lock.json  public  README.md  src
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app$ cat package.json
{
  "name": "my-first-react-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.15.0",
    "@testing-library/react": "^11.2.7",
    "@testing-library/user-event": "^12.8.3",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.1.2"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/4_1_react-nodejs-krishna-rikeshkarma/React JS/my-first-react-app$ 
```

  - We need to change the start script to
    - *"start": "PORT=3001 react-scripts start"*

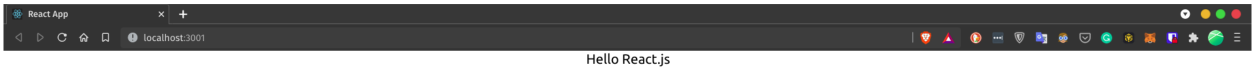


```
  "scripts": {
    "start": "PORT=3001 react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
```

  - Now we can check the port using *npm start.*



```
Compiled successfully!

You can now view my-first-react-app in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.254.141:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

```



Hello React.js