

1. Node JS:

- Install Node js on local VM.
- Create 2 API's running on ports 6080 and 7080 with messages "Hello Node JS" and "Node JS installed successfully" respectively.
- Install pm2 tool and create 4 clusters of both Node's.
- Delete all 4 clusters one-by-one

Answer:

Firstly, we install Node.js and NPM using the NodeSource Enterprise Linux repository. We use following commands;

```
- curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -  
- sudo apt-get install -y nodejs  
- sudo apt-get install npm
```

We use following commands to verify if the Node.js and npm are installed in our system;

```
- node -v  
- npm -v
```

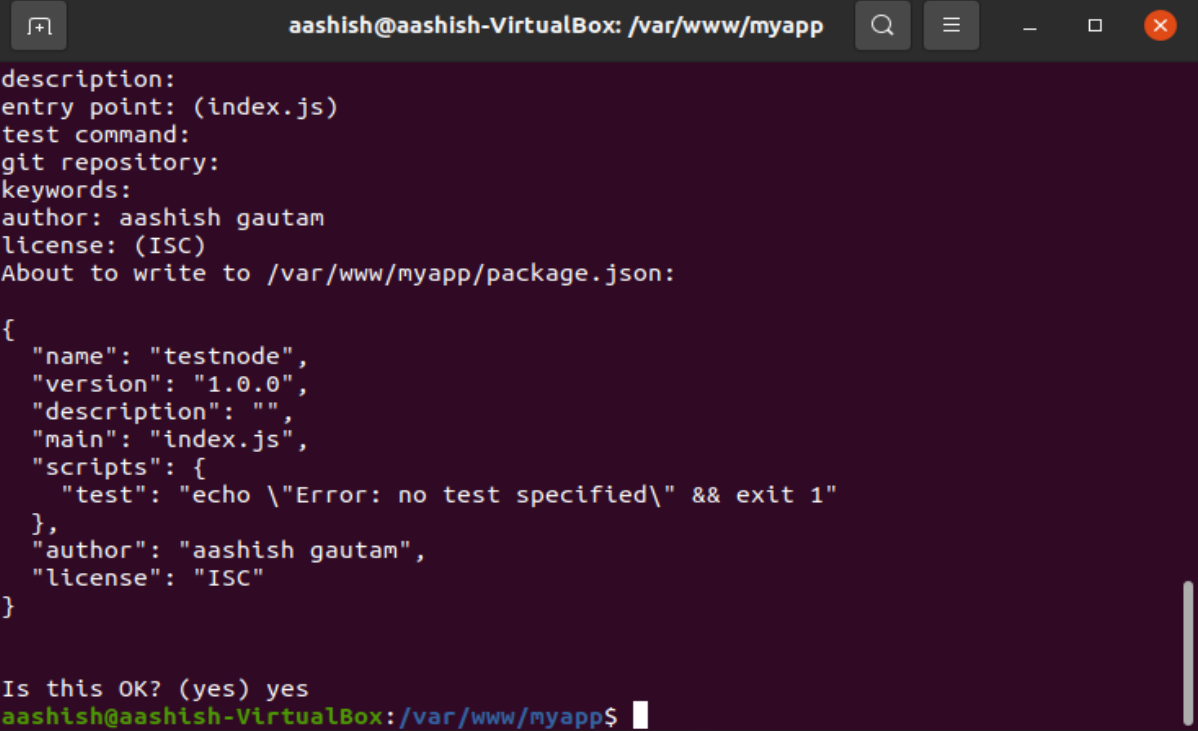
A terminal window titled 'aashish@aashish-VirtualBox: ~' with search and menu icons in the top right. The terminal shows the following commands and output:

```
aashish@aashish-VirtualBox:~$ node -v  
v10.19.0  
aashish@aashish-VirtualBox:~$ npm -v  
6.14.4  
aashish@aashish-VirtualBox:~$
```

Now, we create a directory that will store our application files after initializing npm, using the following command;

```
- sudo mkdir -p /var/www/myapp  
- cd /var/www/myapp  
- npm init
```

And then, app named **testnode** is created as follows;



```
aashish@aashish-VirtualBox: /var/www/myapp
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: aashish gautam
license: (ISC)
About to write to /var/www/myapp/package.json:

{
  "name": "testnode",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aashish gautam",
  "license": "ISC"
}

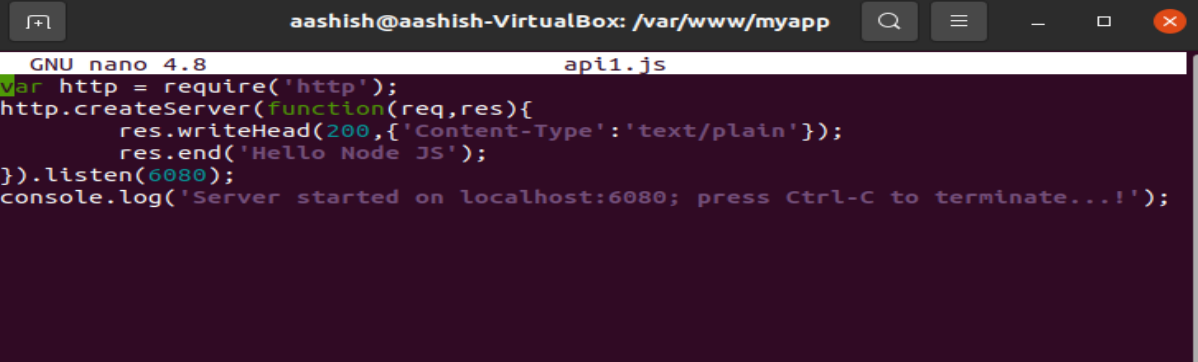
Is this OK? (yes) yes
aashish@aashish-VirtualBox: /var/www/myapp$
```

Next, we create 2 apps namely **api1.js** running on port **6080** and **api2.js** running on port **7080** with messages "Hello Node JS" and "Node JS installed successfully" respectively.

- **sudo nano api1.js**

Then, we copy and paste the code below in it.

```
var http = require('http');
http.createServer(function(req,res){
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('Hello Node JS');
}).listen(6080);
console.log('Server started on localhost:8080; press Ctrl-C to terminate...!');
```

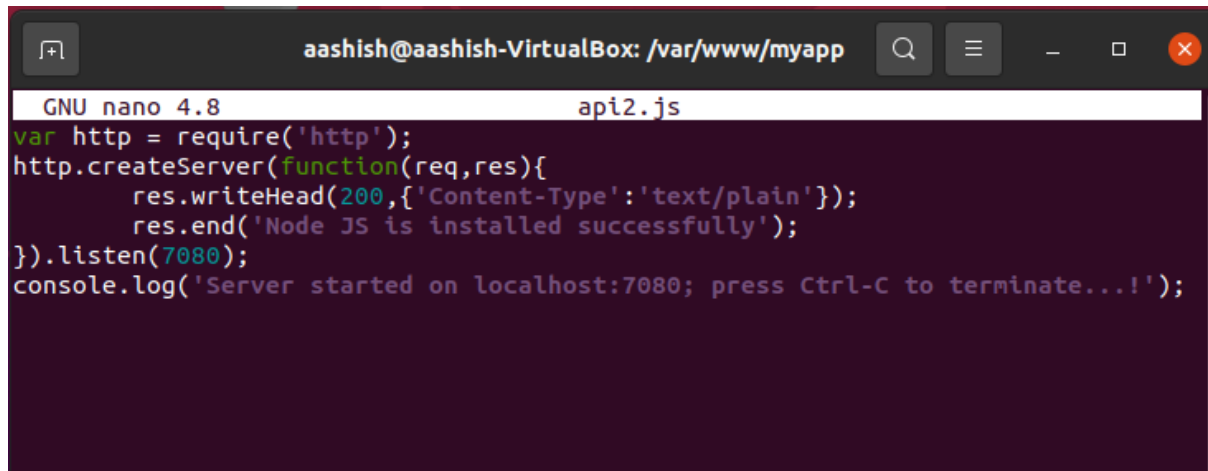


```
GNU nano 4.8 api1.js
var http = require('http');
http.createServer(function(req,res){
    res.writeHead(200,{ 'Content-Type': 'text/plain' });
    res.end('Hello Node JS');
}).listen(6080);
console.log('Server started on localhost:6080; press Ctrl-C to terminate...!');
```

- **sudo nano api2.js**

Again, we copy and paste the code below in it.

```
var http = require('http');
http.createServer(function(req,res){
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('Node JS installed successfully');
}).listen(7080);
console.log('Server started on localhost:8080; press Ctrl-C to terminate...!');
```

A screenshot of a terminal window titled 'aashish@aashish-VirtualBox: /var/www/myapp'. The window shows the GNU nano 4.8 editor editing a file named 'api2.js'. The code in the editor is:

```
var http = require('http');
http.createServer(function(req,res){
    res.writeHead(200,{ 'Content-Type':'text/plain'});
    res.end('Node JS is installed successfully');
}).listen(7080);
console.log('Server started on localhost:7080; press Ctrl-C to terminate...!');
```

Now, to run the app we use following commands;

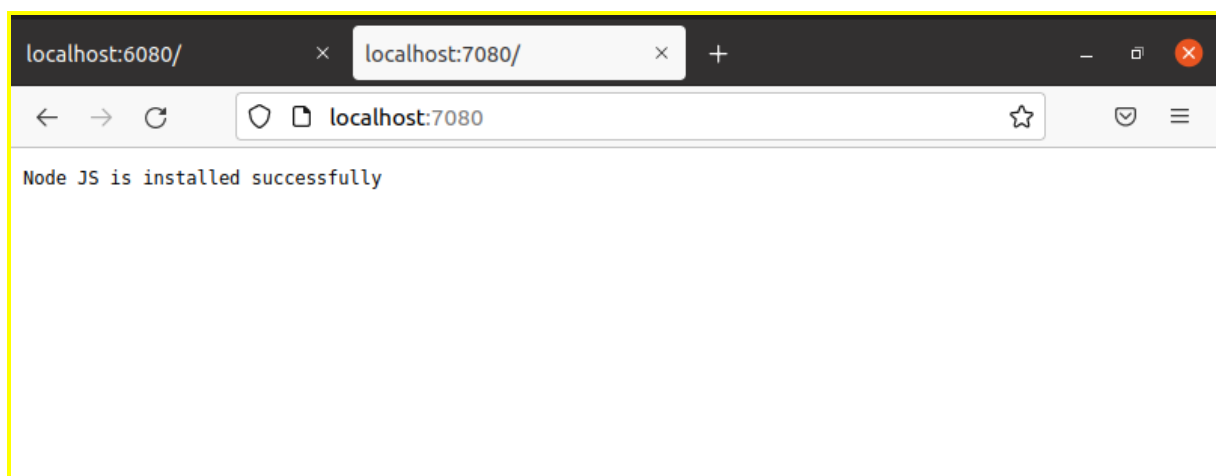
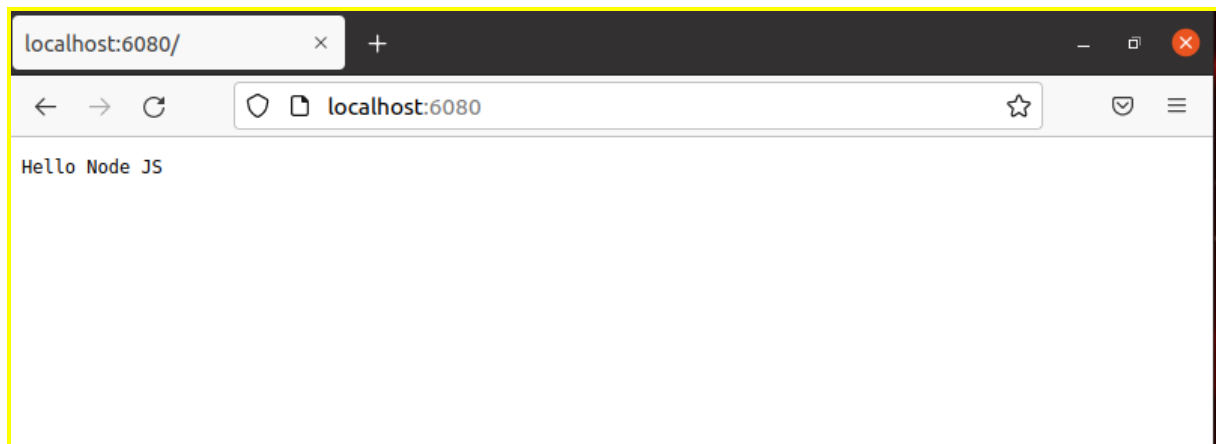
- **node api1.js**
- **node api2.js**

A screenshot of a terminal window titled 'aashish@aashish-VirtualBox: /var/www/myapp'. The terminal shows the following commands and output:

```
aashish@aashish-VirtualBox:/var/www/myapp$ sudo node api1.js
Server started on localhost:6080; press Ctrl-C to terminate...!
^C
aashish@aashish-VirtualBox:/var/www/myapp$ sudo node api2.js
Server started on localhost:7080; press Ctrl-C to terminate...!
```

To test the app, we run following command on the browser;

- **localhost:6080**
- **localhost:7080**



Now, we install PM2 tools using following command;

- sudo npm i -g pm2

```
aashish@aashish-VirtualBox: /var/www/myapp
aashish@aashish-VirtualBox:/var/www/myapp$ sudo npm i -g pm2
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/local/bin/pm2 -> /usr/local/lib/node_modules/pm2/bin/pm2
/usr/local/bin/pm2-dev -> /usr/local/lib/node_modules/pm2/bin/pm2-dev
/usr/local/bin/pm2-docker -> /usr/local/lib/node_modules/pm2/bin/pm2-docker
/usr/local/bin/pm2-runtime -> /usr/local/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 47.268s
```

To verify pm2 tool we use following command;

- **pm2 -v**

```
aashish@aashish-VirtualBox: /var/www/myapp

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/aashish/.pm2
[PM2] PM2 Successfully daemonized
5.1.2
aashish@aashish-VirtualBox: /var/www/myapp$
```

To start node application using pm2 tool, we use following commands;

- **sudo pm2 start api1.js**

- **sudo pm2 start api2.js**

```
aashish@aashish-VirtualBox: /var/www/myapp

aashish@aashish-VirtualBox: /var/www/myapp$ sudo pm2 start api1.js
[PM2] Starting /var/www/myapp/api1.js in fork_mode (1 instance)
[PM2] Done.

  id  name      mode  ⚡  status  cpu  memory
  --  -
  0    api1      fork  0   online  0%   20.5mb

aashish@aashish-VirtualBox: /var/www/myapp$ sudo pm2 start api2.js
[PM2] Starting /var/www/myapp/api2.js in fork_mode (1 instance)
[PM2] Done.

  id  name      mode  ⚡  status  cpu  memory
  --  -
  0    api1      fork  0   online  0%   41.5mb
  1    api2      fork  0   online  0%   15.5mb

aashish@aashish-VirtualBox: /var/www/myapp$
```

To start both applications in 4 cluster mode, we use following command;

- `sudo pm2 start api1.js -i 4`
- `sudo pm2 start api2.js -i 4`

```
aashish@aashish-VirtualBox: /var/www/myapp
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 start api1.js -i 4
[PM2] Starting /var/www/myapp/api1.js in cluster_mode (4 instances)
[PM2] Done.
```

id	name	mode	U	status	cpu	memory
0	api1	cluster	0	online	0%	41.5mb
1	api1	cluster	0	online	0%	42.7mb
2	api1	cluster	0	online	0%	41.4mb
3	api1	cluster	0	online	0%	33.1mb

```
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 start api2.js -i 4
[PM2] Starting /var/www/myapp/api2.js in cluster_mode (4 instances)
[PM2] Done.
```

id	name	mode	U	status	cpu	memory
0	api1	cluster	0	online	0.1%	42.0mb
1	api1	cluster	0	online	0%	43.2mb
2	api1	cluster	0	online	0%	41.9mb
3	api1	cluster	0	online	0%	41.9mb
4	api2	cluster	0	online	0%	41.5mb
5	api2	cluster	0	online	0%	41.6mb
6	api2	cluster	0	online	0%	41.5mb
7	api2	cluster	0	online	0%	33.3mb

To delete all the clusters one-by-one, we use following command;

- `sudo pm2 delete 0`
- `sudo pm2 delete 1`
- `sudo pm2 delete 2`
- `sudo pm2 delete 3`
- `sudo pm2 delete 4`
- `sudo pm2 delete 5`
- `sudo pm2 delete 6`
- `sudo pm2 delete 7`

```
aashish@aashish-VirtualBox: /var/www/myapp
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: 0)
[PM2] [api1](0) ✓
```

id	name	mode	🔄	status	cpu	memory
1	api1	cluster	0	online	0%	45.3mb
2	api1	cluster	0	online	0%	44.0mb
3	api1	cluster	0	online	0%	44.0mb
4	api2	cluster	0	online	0%	44.2mb
5	api2	cluster	0	online	0%	44.5mb
6	api2	cluster	0	online	0%	44.1mb
7	api2	cluster	0	online	0%	45.3mb

```
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 delete 1
[PM2] Applying action deleteProcessId on app [1](ids: 1)
[PM2] [api1](1) ✓
```

id	name	mode	🔄	status	cpu	memory
2	api1	cluster	0	online	0%	44.0mb
3	api1	cluster	0	online	0%	44.0mb
4	api2	cluster	0	online	0%	43.6mb
5	api2	cluster	0	online	0%	44.0mb
6	api2	cluster	0	online	0%	43.5mb
7	api2	cluster	0	online	0%	45.3mb

```
aashish@aashish-VirtualBox:/var/www/myapp$
```

```
aashish@aashish-VirtualBox: /var/www/myapp
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 delete 2
[PM2] Applying action deleteProcessId on app [2](ids: 2)
[PM2] [api1](2) ✓
```

id	name	mode	🔄	status	cpu	memory
3	api1	cluster	0	online	0%	44.0mb
4	api2	cluster	0	online	0%	43.7mb
5	api2	cluster	0	online	0%	44.0mb
6	api2	cluster	0	online	0%	43.6mb
7	api2	cluster	0	online	0%	45.0mb

```
aashish@aashish-VirtualBox:/var/www/myapp$ sudo pm2 delete 3
[PM2] Applying action deleteProcessId on app [3](ids: 3)
[PM2] [api1](3) ✓
```

id	name	mode	🔄	status	cpu	memory
4	api2	cluster	0	online	0%	43.7mb
5	api2	cluster	0	online	0%	44.0mb
6	api2	cluster	0	online	0%	43.6mb
7	api2	cluster	0	online	0%	45.0mb