

---

## **Laboratorio N° 4.4**

### **“Configuración Nginx con Node.js”**

---

## LABORATORIO 4.4

### Objetivos:

- ✓ Habilitar una aplicación con Node.js para trabajarla con pm2.

### INTRODUCCIÓN

Node.js es un marco de JavaScript de código abierto, liviano, escalable y eficiente, construido en el motor JavaScript V8 de Chrome, y utiliza un modelo de E/S sin bloqueo controlado por eventos. Node.js ahora está en todas partes, y se volvió tan popular para desarrollar software desde sitios web, aplicaciones web a aplicaciones de red y más.

### Pasos previos.

#### 1. Agregaremos los últimos paquetes Node.js y NPM.

```
# curl -sL https://rpm.nodesource.com/setup_11.x | bash
# yum install nodejs -y
```

#### 2. Crearemos una aplicación de muestra en una carpeta nueva

```
# mkdir /opt/nodeapp
# cd /opt/nodeapp
```

#### 3. Dentro de esta carpeta iniciaremos la app node.js (aceptar todos los valores predeterminados cuando se le solicite).

```
# npm init
```

#### 4. Instale el paquete express.js.

```
# npm install --save express
```

#### 5. Usar un editor de texto para crear app.js y agregar el siguiente contenido.

```
# vim app.js
const express = require('express')
const app = express()

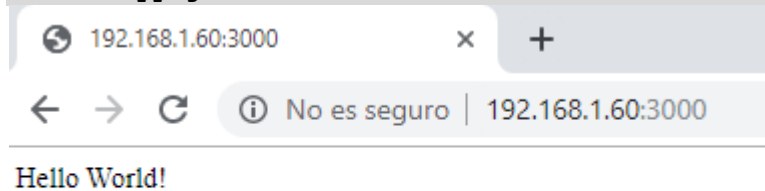
app.get('/', (req, res) => res.send('Hello World!'))

app.listen(3000, () => console.log('Node.js app listening on port 3000.'))
```

---

6. Ejecutaremos la aplicación.

```
# node app.js
```



7. Cree un archivo de configuración para la aplicación REEMPLAZAR LA IP.

```
# vim /etc/nginx/conf.d/sysmon.conf
server {
    listen 80;
    server_name 192.168.1.60;

    location / {
        proxy_set_header    X-Forwarded-For $remote_addr;
        proxy_set_header    Host $http_host;
        proxy_pass            http://192.168.1.60:3000;
    }
}
```

8. Reemplace o elimine la página predeterminada BIENVENIDO a NGINX.

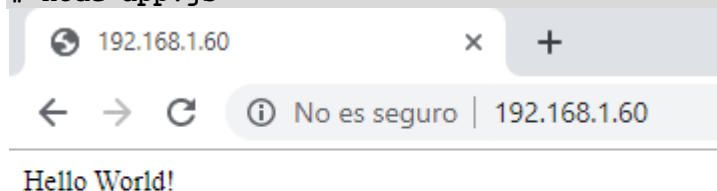
```
# mv /etc/nginx/conf.d/default.conf
/etc/nginx/conf.d/default.conf.backup
```

9. Probar la configuración.

```
# nginx -t
```

10. Si no se informan errores, vuelva a cargar la nueva configuración.

```
# nginx -s reload
# node app.js
```



11. Utilizaremos los módulos empaquetados de nodo (NPM), que es básicamente un administrador de paquetes para módulos de nodos que se instala con Nodejs, para instalar PM2 en nuestro servidor de aplicaciones. Use este comando para instalar PM2

```
# npm install pm2 -g
[root@srvweb nodeapp]# npm install pm2 -g
/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-runtime -> /usr/lib/node_modules/pm2/bin/pm2-runtime
/usr/bin/pm2-docker -> /usr/lib/node_modules/pm2/bin/pm2-docker
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/pm2/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
+ pm2@8.3.5.1
added 319 packages from 258 contributors in 13.505s
[root@srvweb nodeapp]#
```

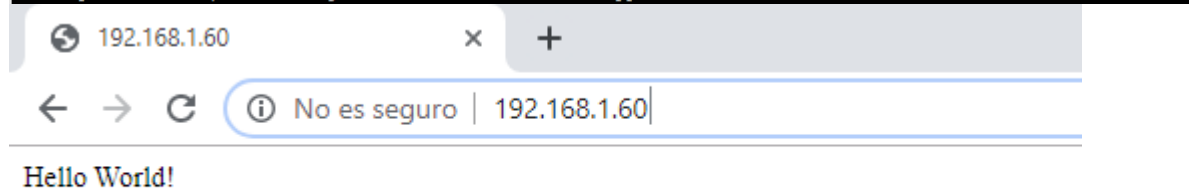
12. Lo primero que querrá hacer es usar el comando pm2 start para ejecutar su aplicación con el nombre de la aplicación hola-mundo.

```
# pm2 start app.js --name hola-mundo

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /root/nodeapp/app.js in fork_mode (1 instance)
[PM2] Done.
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
hola-mundo	0	1.0.0	fork	1783	online	0	0s	0%	13.2 MB	root	disabled

Use 'pm2 show <id|name>' to get more details about an app



13. Modificaremos el archivo app.js (Hello World por Hola Mundo).

```
# vim app.js
```

14. Reiniciar la aplicación.

```
# pm2 restart app.js

[PM2] Applying action restartProcessId on app [app.js] (ids: 0)
[PM2] [hola-mundo] (0) ✓
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
hola-mundo	0	1.0.0	fork	1816	online	1	0s	0%	10.4 MB	root	disabled

Use 'pm2 show <id|name>' to get more details about an app

