Iniciamos pm2 en un archivo node.

Ejecutamos la linea de comando que nos proporciona para poder ejecutar pm2 como superusuario.

```
ivan@vps591614:/$ sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u ivan --hp /home/ivan/7
env: '/usr/lib/node_modules/pm2/bin/pm2': No such file or directory
ivan@vps591614:/$ sudo env PATH=$PATH:/usr/local/bin /usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u ivan --hp /home/ivan/7
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=ivan
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/local/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/home/ivan/7/.pm2
PIDFile=/home/ivan/7/.pm2/pm2.pid

ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/local/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/local/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-ivan.service
Command list
[ 'systemctl enable pm2-ivan' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-ivan.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-ivan...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-ivan.service → /etc/systemd/system/pm2-ivan.service.
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
ivan@vps591614:/$
```

Pm2save `sudo systemctl start pm2-ivan`

```
ivan@vps591614:/$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ivan/.pm2/dump.pm2
ivan@vps591614:/$ sudo systemctl start pm2-ivan
ivan@vps591614:/$ systemctl status pm2-ivan
● pm2-ivan.service - PM2 process manager
   Loaded: loaded (/etc/systemd/system/pm2-ivan.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-01-17 18:45:00 CET; 10s ago
     Docs: https://pm2.keymetrics.io/
  Process: 27979 ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect (code=exited, status=0/SUCCESS)
 Main PID: 27986 (PM2 v3.2.9: God)
    Tasks: 11 (limit: 4915)
   CGroup: /system.slice/pm2-ivan.service
           └─27986 PM2 v3.2.9: God Daemon (/home/ivan/7/.pm2)
ivan@vps591614:/$
```

Paramos el proceso

```
ivan@vps591614:/$ pm2 stop app
[PM2] Applying action stopProcessId on app [app](ids: 0)
[PM2] [app](0) ✓
```

Name	id	mode	status	□	cpu	memory	
app	0	0.0.0	fork	stopped	0	0%	0 B

```
 Use `pm2 show <id|name>` to get more details about an app
ivan@vps591614:/$
```

Reiniciamos el proceso

```
ivan@vps591614:/$ pm2 restart app
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [app](ids: 0)
[PM2] [app](0) ✓
```

Name	id	mode	status	□	cpu	memory	
app	0	0.0.0	fork	online	0	0%	4.9 MB

```
 Use `pm2 show <id|name>` to get more details about an app
ivan@vps591614:/$
```

Listamos todos los procesos pm2 list

```
ivan@vps591614:/$ pm2 list
```

Name	id	mode	status	□	cpu	memory	
app	0	0.0.0	fork	online	0	0%	46.8 MB

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

Pm2 info

```
ivan@vps591614:/$ pm2 info app
 Describing process with id 0 - name app
```

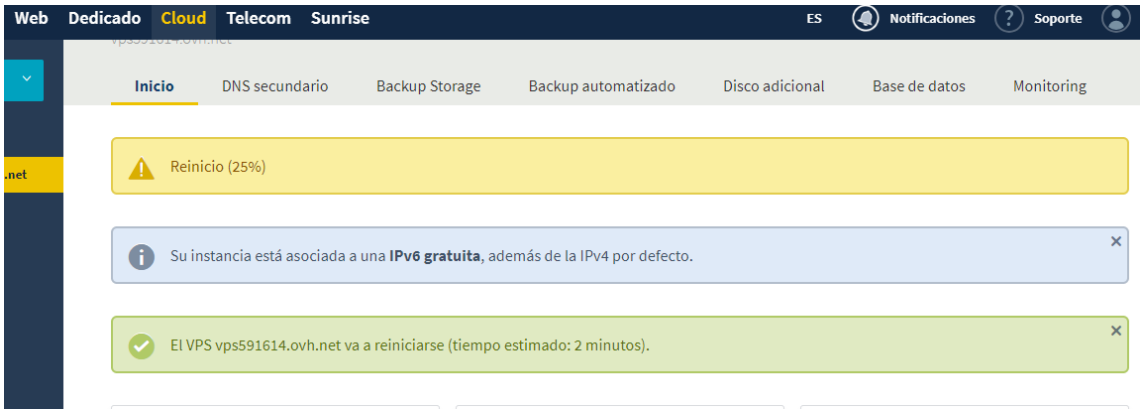
status	online
name	app
version	0.0.0
restarts	0
uptime	63s
script path	/home/ivan/proyecto_dpl/myapp/app.js
script args	N/A
error log path	/home/ivan/.pm2/logs/app-error.log
out log path	/home/ivan/.pm2/logs/app-out.log
pid path	/home/ivan/.pm2/pids/app-0.pid
interpreter	node
interpreter args	N/A
script id	0
exec cwd	/
exec mode	fork_mode
node.js version	10.10.0
node env	N/A
watch & reload	✗
unstable restarts	0
created at	2019-01-17T17:50:33.083Z

```
 Code metrics value
```

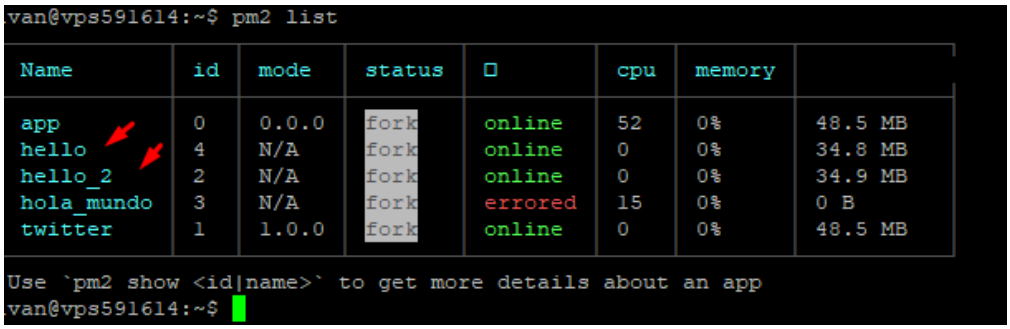
Event Loop Latency	1.18ms
Active handles	6

```
 Add your own code metrics: http://bit.ly/code-metrics
 Use `pm2 logs app [--lines 1000]` to display logs
 Use `pm2 env 0` to display environement variables
 Use `pm2 monit` to monitor CPU and Memory usage app
ivan@vps591614:/$
```

Procedemos a realizar un reinicio del servidor para ver si la aplicación arranca como un servicio.

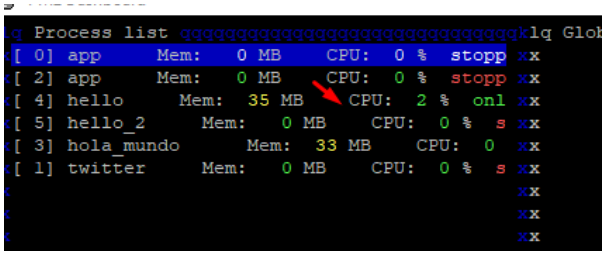


Una vez reiniciado realizamos un pm2 list y comprobamos como estan activos.

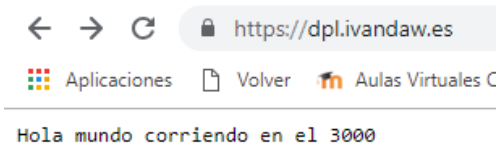


Ejecutamos pm2 monit

Observamos el monitoreo en tiempo real de nuestros procesos.



Configuramos el server block y probamos que la apliación se ejecuta correctamente



Arrancamos otro proceso (hello_2)

```
[PM2] Process successfully started
```

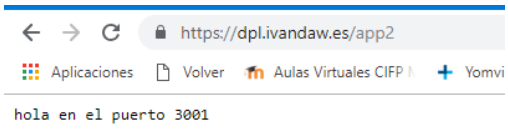
App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
app	0	0.0.0	fork	0	stopped	0	0	0%	0 B	ivan	disabled
app	2	0.0.0	fork	0	stopped	0	0	0%	0 B	ivan	disabled
hello	4	N/A	fork	29689	online	0	4m	0%	32.9 MB	ivan	disabled
hello_2	5	N/A	fork	0	stopped	0	0	0%	0 B	ivan	disabled
hola_mundo	3	N/A	fork	29760	online	0	0s	0%	7.1 MB	ivan	disabled
twitter	1	1.0.0	fork	0	stopped	39	0	0%	0 B	ivan	disabled

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

Agregamos otro Location (app2).

```
server {

        root /var/www/example.com/html;
        index index.html index.htm index.nginx-debian.html;

        server_name dpl.ivandaw.es www.ivandaw.es;

         location / {
#                try_files $uri $uri/ =404;
                proxy_pass http://51.254.116.159:3000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }

        location /app2 {
#                try_files $uri $uri/ =404;
                proxy_pass http://51.254.116.159:3001;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }
```

Ponemos /app2 , y vemos que funciona correctamente




```
hola en el puerto 3001
```

Comprobamos los puertos que están en escucha (3000,3001) en IPv6

```
ivan@vps591614:~$ netstat -ntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3000                 :::*                    LISTEN
tcp6       0      0 :::3001                 :::*                    LISTEN
tcp6       0      0 :::443                  :::*                    LISTEN
ivan@vps591614:~$
```

Ejercicio Node en el puerto 80

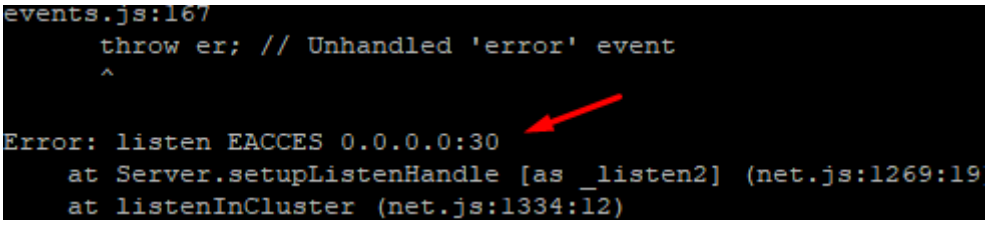
1. Cambia tu aplicación node para que escuche en el puerto 80. ¿Qué sucede?

   Ocurre un error ya que otra aplicación (nginx) está trabajando con el puerto 80.

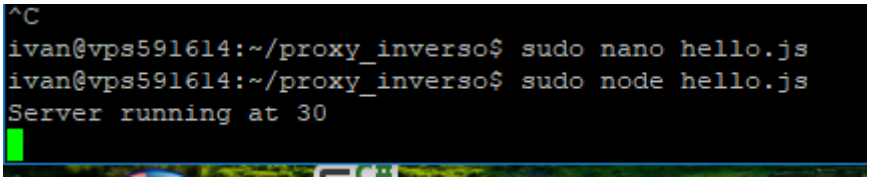
   Además no se pueden asignar puertos por debajo de 1024 como usuarios normales.

2. Ejecuta la aplicación como root y comprueba que funciona.

   Asigne el puerto 30, ejecutamos como usuario normal vemos que no deja.



   Ejecutamos como root



3. ¿Es conveniente esta configuración?

No es conveniente, al ser ejecutada con privilegios de administrador cualquier fallo o agujero de seguridad en la aplicación sería perjudicial para el servidor, ya que podrían manejar por consola el sistema con privilegios ROOT. Teniendo así acceso a los archivos más críticos del sistema.