Desafío Clusters y escalabilidad:

Usando libreria cluster:

Para este caso se debe requerir la librería y usar el siguiente código:

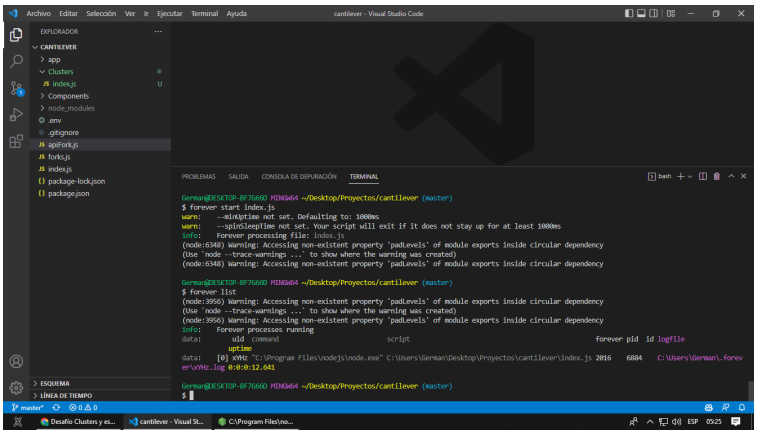
```
const cluster = require('cluster')
const http = require('http')

const numCPUs = require('os').cpus().length


if (cluster.isPrimary) {
  console.log(`Proceso primario: ${process.pid}`)

  for (let i= 0; i < numCPUs; i++) {
    cluster.fork()
  }

  cluster.on('exit', (worker, code, signal) => {
    console.log({ worker, code, signal })
  })
} else {
  console.log(`Proceso worker: ${process.pid}`)

  const PORT = process.argv[2] || 8080

  http.createServer((req, res) => {
    res.writeHead(200)
    res.end(`Process: ${process.pid}`)
  }).listen(PORT)
}
```

Usando Forever:

Para iniciar en modo fork se debe usar el siguiente comando:

```
forever start index.js
```

Para ver los archivos iniciados se usa forever list que da como resultado:

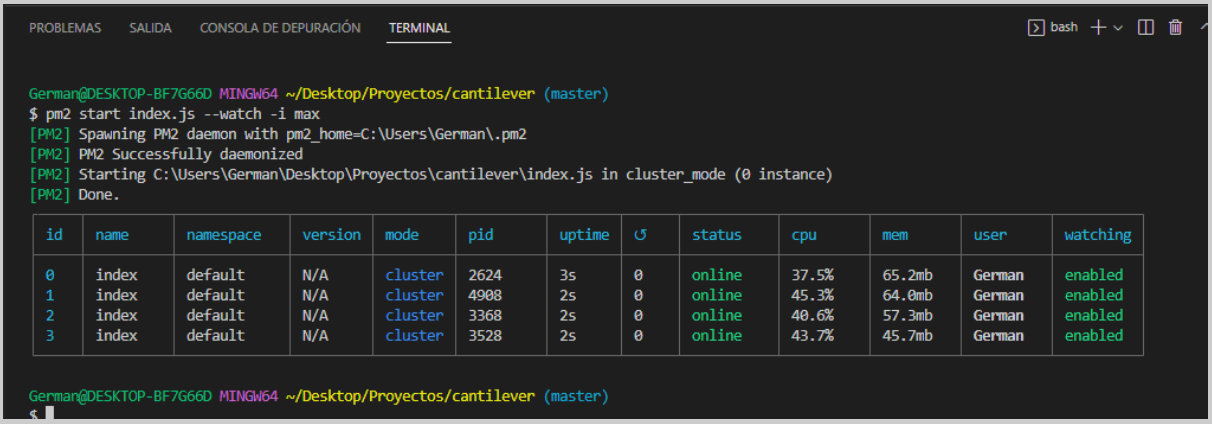


Para detener el proceso se utiliza forever stop {pid}

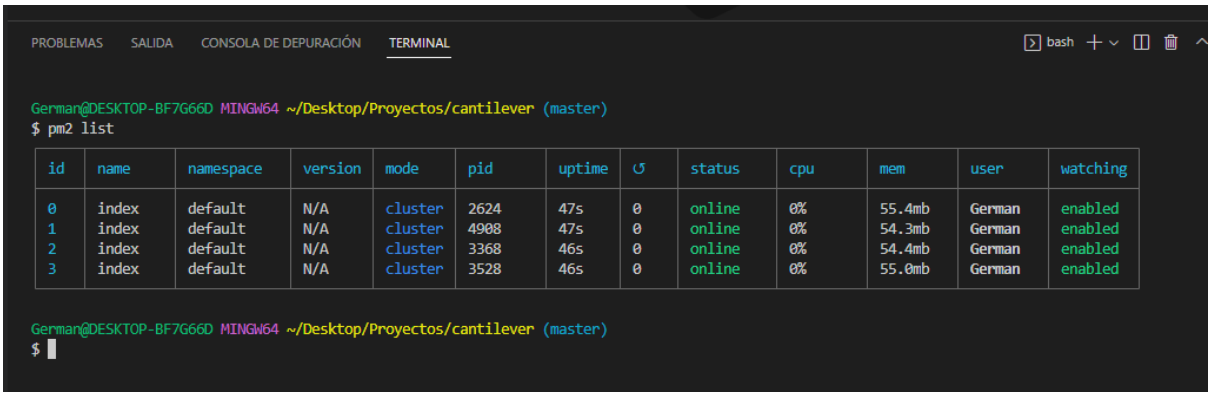
Usando pm2 modo cluster:

Se debe ejecutar el siguiente comando:

```
pm2 start index.js  —-watch -i max
```

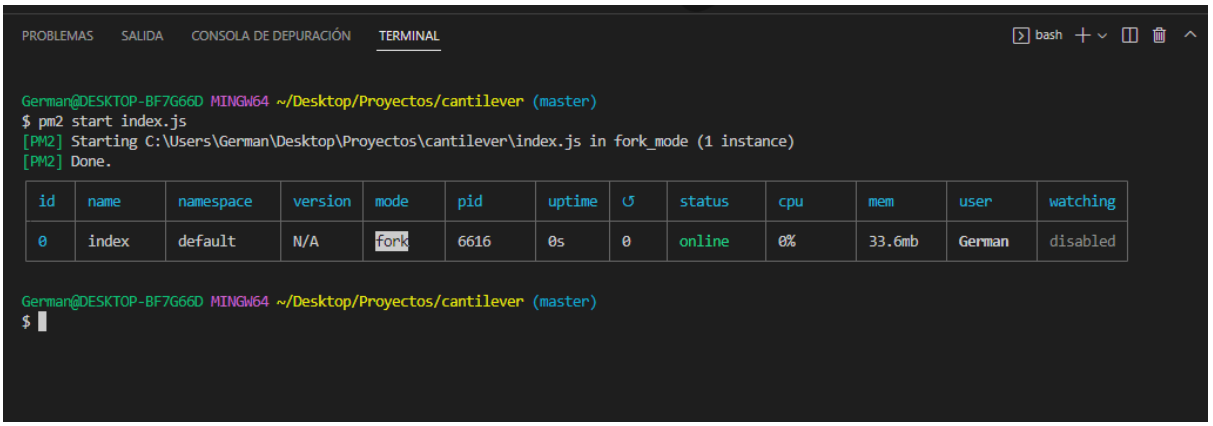


y lo podemos comprobar con pm2 list:



Usando pm2 modo fork:

Se utiliza el comando pm2 start index.js



Nginx como balanceador de carga:

Redirigir todas las consultas a /api/random a un cluster de servidores con la librería clusters:

Para eso de debe modificar /etc/nginx/sites-available/default y agregar lo siguiente:

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
```

```
#
upstream node-app {
	server 172.0.0.1:8081;
}

server {
	listen 80 default_server;
	listen [::]:80 default_server;

	# SSL configuration
	#
	# listen 443 ssl default_server;
	# listen [::]:443 ssl default_server;
	#
	# Note: You should disable gzip for SSL traffic.
	# See: https://bugs.debian.org/773332
	#
	# Read up on ssl_ciphers to ensure a secure configuration.
	# See: https://bugs.debian.org/765782
	#
	# Self signed certs generated by the ssl-cert package
	# Don't use them in a production server!
	#
	# include snippets/snakeoil.conf;

	root /var/www/html;

	# Add index.php to the list if you are using PHP
	index index.html index.htm index.nginx-debian.html;

	server_name _;

	location / {
		# First attempt to serve request as file, then
		# as directory, then fall back to displaying a 404.
		try_files $uri $uri/ =404;
	}

	location /api/random/ {proxy_pass http://node-app/};


	# pass PHP scripts to FastCGI server
	#
	#location ~ \.php$ {
	#	include snippets/fastcgi-php.conf;
	#
	#	# With php-fpm (or other unix sockets):
	#	fastcgi_pass unix:/run/php/php7.4-fpm.sock;
	#	# With php-cgi (or other tcp sockets):
	#	fastcgi_pass 127.0.0.1:9000;
	#}

	# deny access to .htaccess files, if Apache's document root
	# concurs with nginx's one
```

```
        #
        #location ~ /\.ht {
        #       deny all;
        #}
}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#       listen 80;
#       listen [::]:80;
#
#       server_name example.com;
#
#       root /var/www/example.com;
#       index index.html;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}
```

Luego de esto reiniciar nginx con el comando:

```
sudo systemctl restart nginx
```

Configurar Nginx para que las consultas vayan a un cluster de 4 servidores: