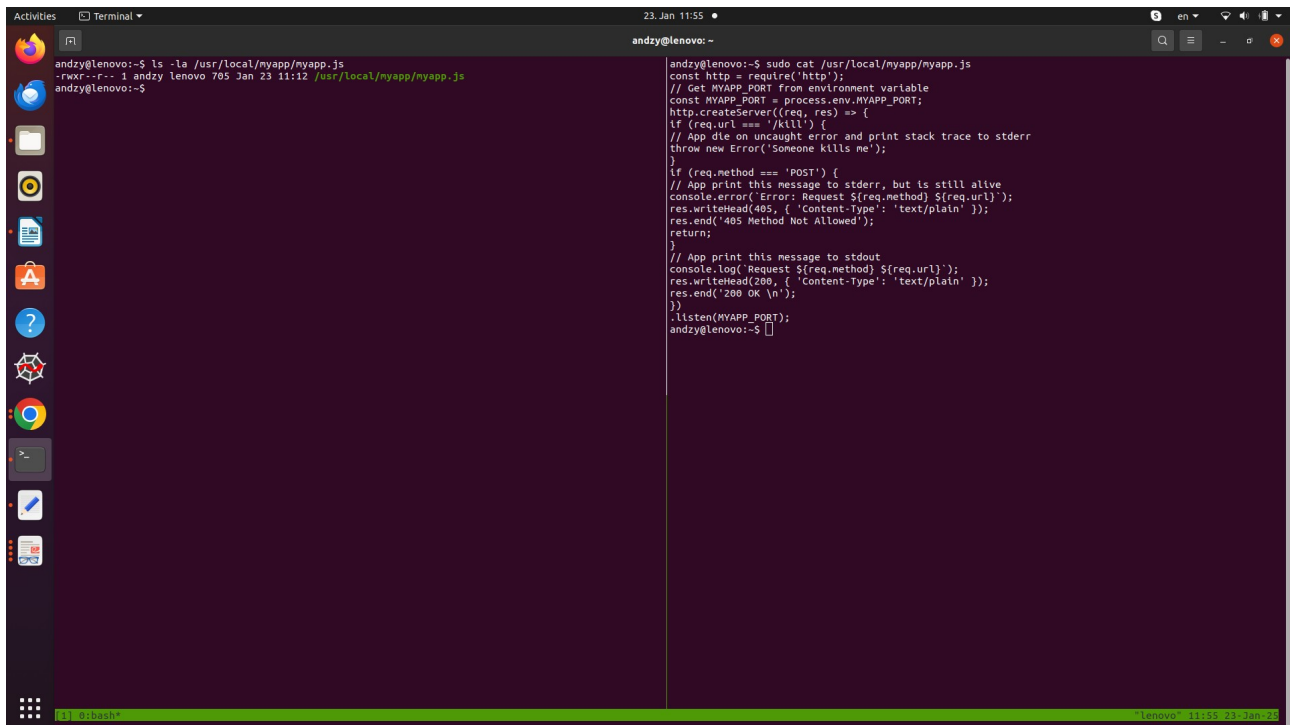


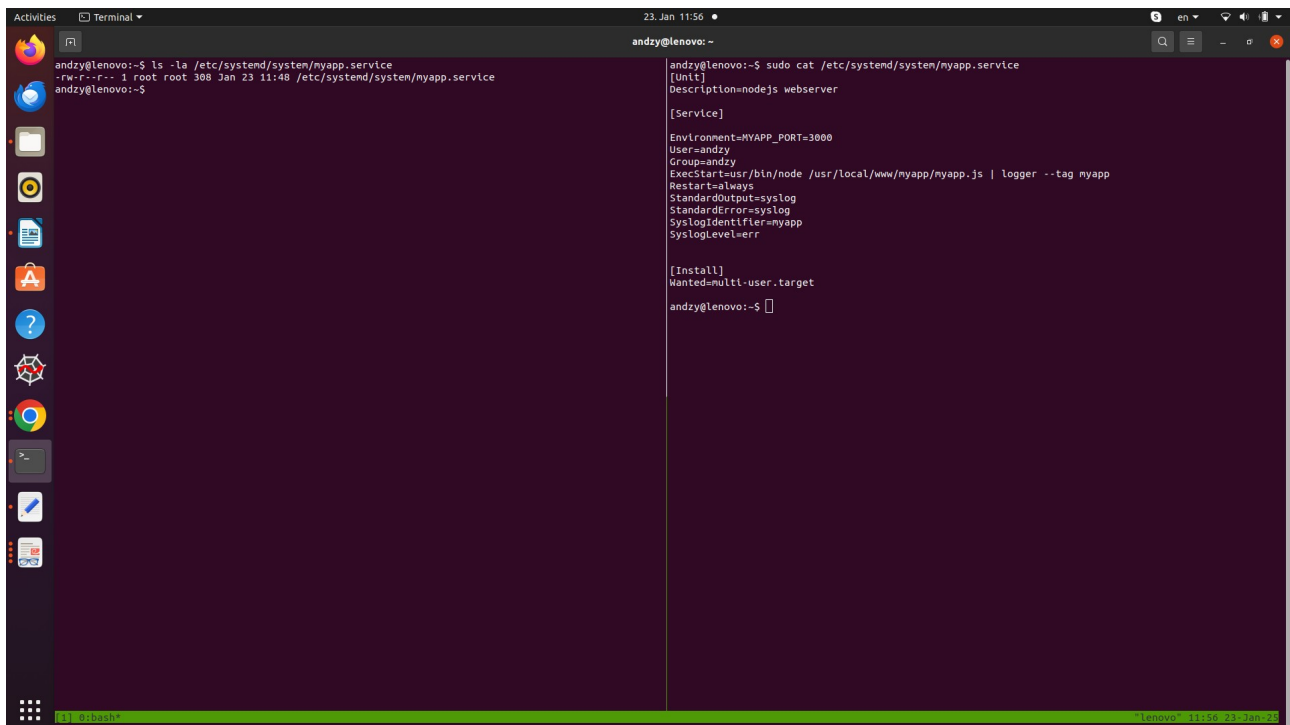
1. Создаем приложение `/usr/local/myapp/myapp.js`

A terminal window on a Linux system. The user 'andy' is logged in as 'andzy' on a machine named 'lenovo'. The terminal shows the user navigating to the directory `/usr/local/myapp` and creating a file `myapp.js` with the `cat` command. The file contains a Node.js script that listens on `MYAPP_PORT` (defaulting to 3000). It handles a `kill` signal by throwing an error, and a `POST` request by returning a 405 status. It also logs requests to stdout and returns a 200 status for other requests.

```
andzy@lenovo:~$ ls -la /usr/local/myapp/myapp.js
-rwxr--r-- 1 andzy lenovo 705 Jan 23 11:12 /usr/local/myapp/myapp.js
andzy@lenovo:~$

andzy@lenovo:~$ sudo cat /usr/local/myapp/myapp.js
const http = require('http');
// Get MYAPP_PORT from environment variable
const MYAPP_PORT = process.env.MYAPP_PORT;
http.createServer((req, res) => {
  if (req.url === '/kill') {
    // App die on uncaught error and print stack trace to stderr
    throw new Error('Someone kills me');
  }
  if (req.method === 'POST') {
    // App print this message to stderr, but is still alive
    console.error('Error: Request ${req.method} ${req.url}');
    res.writeHead(405, { 'Content-Type': 'text/plain' });
    res.end('405 Method Not Allowed');
    return;
  }
  // App print this message to stdout
  console.log('Request ${req.method} ${req.url}');
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('200 OK \n');
}).listen(MYAPP_PORT);
andzy@lenovo:~$
```

2. Создаем сервис-юнит `/etc/systemd/system/myapp.service`

A terminal window on a Linux system. The user 'andy' is logged in as 'andzy' on a machine named 'lenovo'. The terminal shows the user navigating to the directory `/etc/systemd/system` and creating a file `myapp.service` with the `cat` command. The file contains a systemd service unit for a Node.js webserver. It sets the environment variable `MYAPP_PORT=3000`, runs the service as user 'andzy' and group 'andzy', and starts it with `usr/bin/node /usr/local/www/myapp/myapp.js | logger --tag myapp`. It also sets `Restart=always`, `StandardOutput=syslog`, `StandardError=syslog`, `SyslogIdentifier=myapp`, and `SyslogLevel=err`. The `[Install]` section sets `Wanted=multi-user.target`.

```
andzy@lenovo:~$ ls -la /etc/systemd/system/myapp.service
-rw-r--r-- 1 root root 388 Jan 23 11:48 /etc/systemd/system/myapp.service
andzy@lenovo:~$

andzy@lenovo:~$ sudo cat /etc/systemd/system/myapp.service
[Unit]
Description=nodejs webserver

[Service]

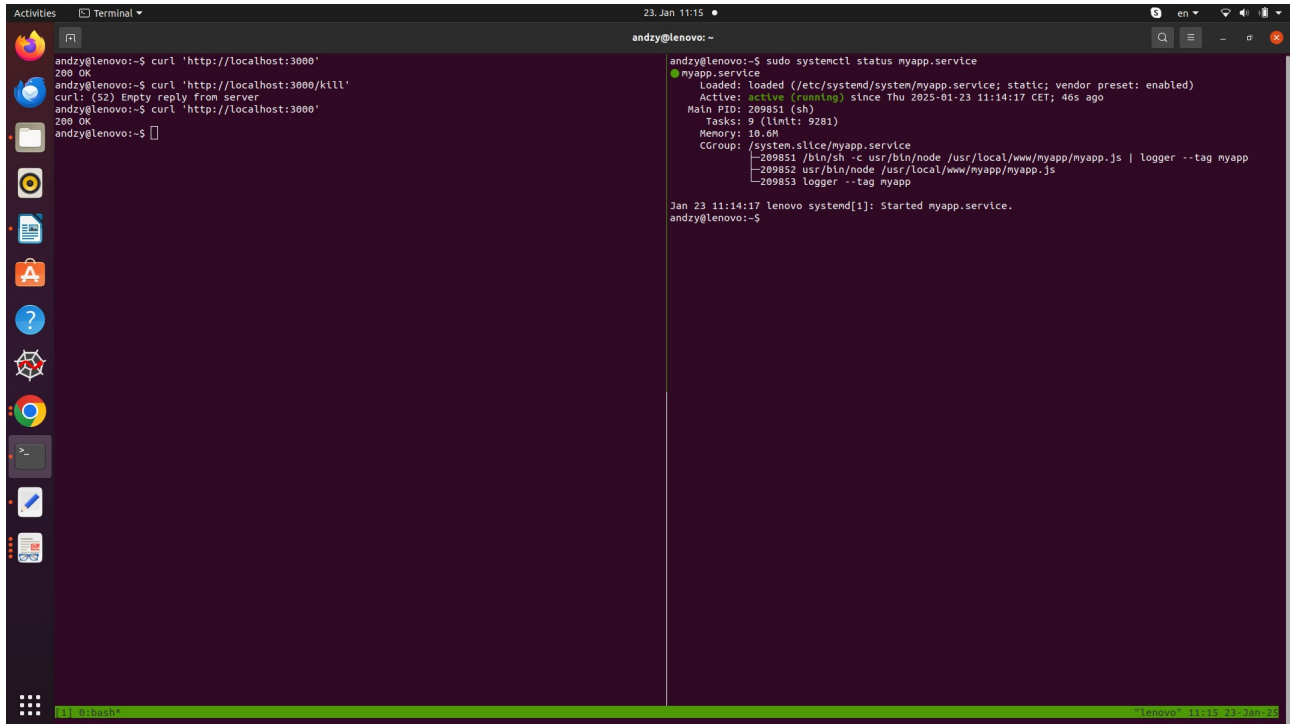
Environment=MYAPP_PORT=3000
User=andzy
Group=andzy
ExecStart=usr/bin/node /usr/local/www/myapp/myapp.js | logger --tag myapp
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=myapp
SyslogLevel=err

[Install]
Wanted=multi-user.target
andzy@lenovo:~$
```

3. Перезапускаем daemon-reload, проверяем статус сервиса и запускаем приложение с проверкой автоматического перезапуска.

```
$ sudo systemctl daemon-reload
$ sudo systemctl start myapp.service
$ sudo systemctl status myapp.service
$ curl 'http://localhost:3000'
$ curl 'http://localhost:3000/kill'
```

Приложение перезапускается после отправки процессу сигнала <KILL>



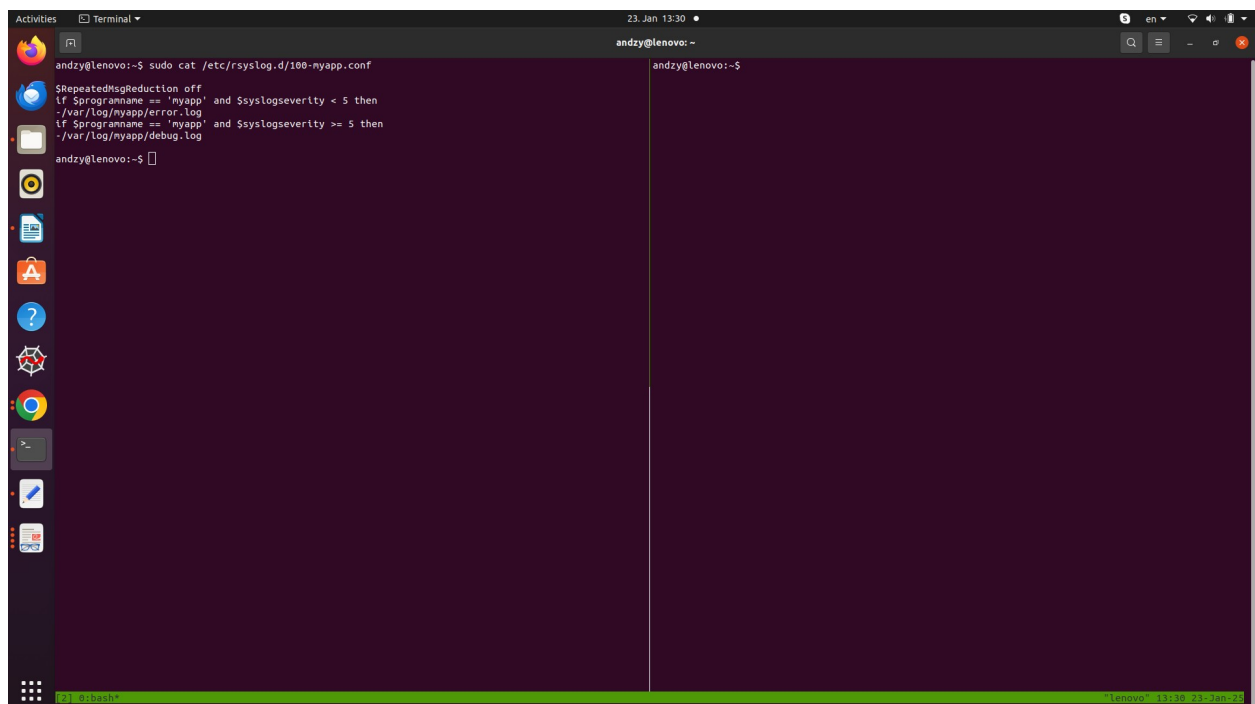
```
andzy@lenovo:~$ curl 'http://localhost:3000'
200 OK
andzy@lenovo:~$ curl 'http://localhost:3000/kill'
curl: (52) Empty reply from server
andzy@lenovo:~$ curl 'http://localhost:3000'
200 OK
andzy@lenovo:~$

andzy@lenovo:~$ sudo systemctl status myapp.service
● myapp.service
   Loaded: loaded (/etc/systemd/system/myapp.service; static; vendor preset: enabled)
   Active: active (running) since Thu 2025-01-23 11:14:17 CET; 46s ago
     Main PID: 209851 (sh)
       Tasks: 9 (limit: 9281)
      Memory: 10.6M
      CGroup: /system.slice/myapp.service
              └─209851 /bin/sh -c usr/bin/node /usr/local/www/myapp/myapp.js | logger --tag myapp
                  └─209852 usr/bin/node /usr/local/www/myapp/myapp.js
                      └─209853 logger --tag myapp

Jan 23 11:14:17 lenovo systemd[1]: Started myapp.service.
andzy@lenovo:~$
```

4. Логирование сервиса.

Создаем файл “/etc/rsyslog.d/100-myapp.conf “ и настраиваем правила для записи в файлы “error.log” и “debug.log”.

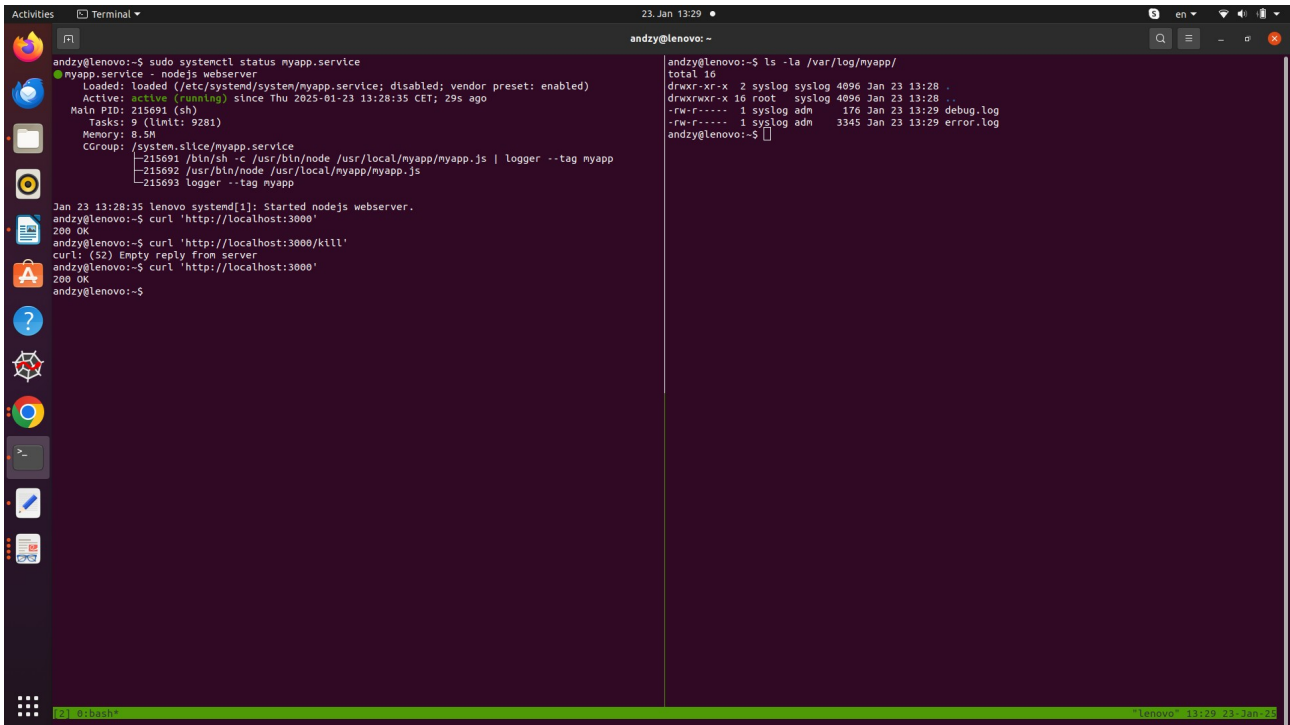


```
andzy@lenovo:~$ sudo cat /etc/rsyslog.d/100-myapp.conf
$RepeatedMsgReduction off
if $programname == 'myapp' and $syslogseverity < 5 then
    /var/log/myapp/error.log
if $programname == 'myapp' and $syslogseverity >= 5 then
    /var/log/myapp/debug.log
andzy@lenovo:~$
```

Перезапускаем службу лог-агрегатор syslog

```
$ sudo systemctl daemon-reload  
$ sudo systemctl restart syslog
```

Теперь логи записываются в файлы `var/log/myapp/debug.log` и `var/log/myapp/error.log`



```
andzy@lenovo:~$ sudo systemctl status myapp.service  
● myapp.service - nodejs webserver  
   Loaded: loaded (/etc/systemd/system/myapp.service; disabled; vendor preset: enabled)  
   Active: active (running) since Thu 2025-01-23 13:28:35 CET; 29s ago  
     Main PID: 215691 (sh)  
       Tasks: 9 (limit: 9281)  
      Memory: 8.5M  
     CGroup: /system.slice/myapp.service  
             └─215691 /bin/sh -c /usr/bin/node /usr/local/myapp/myapp.js | logger --tag myapp  
               └─215692 /usr/bin/node /usr/local/myapp/myapp.js  
                 └─215693 logger --tag myapp  
  
Jan 23 13:28:35 lenovo systemd[1]: Started nodejs webserver.  
andzy@lenovo:~$ curl 'http://localhost:3000'  
200 OK  
andzy@lenovo:~$ curl 'http://localhost:3000/kill'  
curl: (52) Empty reply from server  
andzy@lenovo:~$ curl 'http://localhost:3000'  
200 OK  
andzy@lenovo:~$  
  
andzy@lenovo:~$ ls -la /var/log/myapp/  
total 16  
drwxr-xr-x  2 syslog syslog 4096 Jan 23 13:28 .  
drwxrwxr-x 16 root    syslog 4096 Jan 23 13:28 ..  
-rw-r----- 1 syslog adm   176 Jan 23 13:29 debug.log  
-rw-r----- 1 syslog adm  3345 Jan 23 13:29 error.log  
andzy@lenovo:~$
```