

Цель: получить практический опыт установки пакетов с помощью сторонних репозиториев и пакетного менеджера. Научиться выполнять задачи автоматизации с помощью Bash.

Задание 1 – добавить в cron скрипт/команду, которая будет очищать кэш apt (кэшируемые пакеты, пакеты, которые не могут быть загружены) раз в месяц в 16 часов.

Опционально:

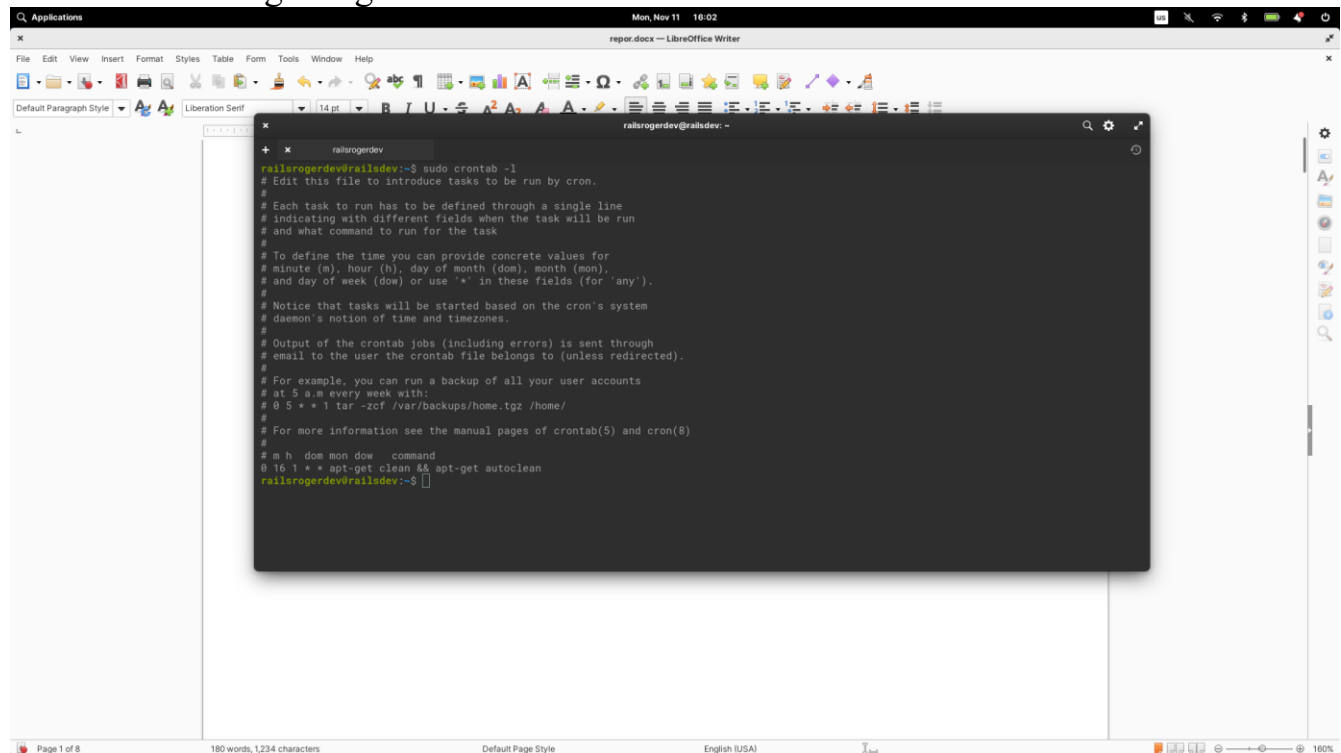
Задание 2 – в соответствии с нижеуказанным примером, запустить демон nodejs-приложения через systemd.

TASK 1

1. Update packages and install cron. Enable it to work in the background:

`sudo systemctl enable cron`

We edit the config using the command: `sudo crontab -e`



TASK 2

1. Install NodeJS.
2. Create scripts and testing.

The screenshot shows the Visual Studio Code editor with a file named `index.js` open. The code is a simple HTTP server using the `http` module. It listens on `MYAPP_PORT` (which is `3000` in the terminal). The server has two endpoints: a `POST` endpoint that returns a 405 status code and a message, and a `GET` endpoint that returns a 200 status code and the message "200 OK".

```
1 const http = require('http');
2 // Get MYAPP_PORT from environment variable
3 const MYAPP_PORT = process.env.MYAPP_PORT;
4 http.createServer((req, res) => {
5   if (req.url === '/kill') {
6     // App die on uncaught error and print stack trace to stderr
7     throw new Error('Someone kills me');
8   }
9   if (req.method === 'POST') {
10    // App print this message to stderr, but is still alive
11    console.error('Error: Request ${req.method} ${req.url}');
12    res.writeHead(405, { 'Content-Type': 'text/plain' });
13    res.end('405 Method Not Allowed');
14    return;
15  }
16  // App print this message to stdout
17  console.log('Request ${req.method} ${req.url}');
18  res.writeHead(200, { 'Content-Type': 'text/plain' });
19  res.end('200 OK');
20 })
21 .listen(MYAPP_PORT);
```

The terminal window shows the command `railsrogerdev@railsdev: ~/MA_DOS24/Homework_Lesson5_OS_4` and the execution of `node index.js`. The output shows the server listening on port 3000 and receiving several requests. The first three requests are `GET` requests to `http://localhost:3000`, and the output shows `200 OK` for each.

```
railsrogerdev@railsdev: ~/MA_DOS24/Homework_Lesson5_OS_4
railsrogerdev@railsdev:~/MA_DOS24/Homework_Lesson5_OS_4$ MYAPP_PORT=3000 node index.js
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
```

Sending a GET request

```
index.js
1 const http = require('http');
2 // Get MYAPP_PORT from environment variable
3 const MYAPP_PORT = process.env.MYAPP_PORT;
4 http.createServer((req, res) => {
5   if (req.url === '/kill') {
6     // App die on uncaught error and print stack trace to stderr
7     throw new Error('Someone kills me');
8   }
9   if (req.method === 'POST') {
10    // App print this message to stderr, but is still alive
11    console.error('Error: Request ${req.method} ${req.url}');
12    res.writeHead(405, { 'Content-Type': 'text/plain' });
13    res.end('405 Method Not Allowed');
14    return;
15  }
16  // App print this message to stdout
17  console.log('Request: ${req.method} ${req.url}');
18}
```

```
railsrogerdev@railsdev: ~/MA_DOS24/Homework_Lesson5_OS_4
railsrogerdev@railsdev:~/MA_DOS24/Homework_Lesson5_OS_4$ MYAPP_PORT=3000 node index.js
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Request ${req.method} ${req.url}
Error: Request ${req.method} ${req.url}
/home/railsrogerdev/MA_DOS24/Homework_Lesson5_OS_4/index.js:7
    throw new Error('Someone kills me');
    ^
Error: Someone kills me
    at Server.<anonymous> (/home/railsrogerdev/MA_DOS24/Homework_Lesson5_OS_4/index.js:7:15)
    at Server.emit (node:events:587:28)
    at parserOnIncoming (node:_http_server:1146:12)
    at HTTPParser.parserOnHeadersComplete (node:_http_common:117:17)
Node.js v23.1.0
railsrogerdev@railsdev:~/MA_DOS24/Homework_Lesson5_OS_4$
```

```
railsrogerdev@railsdev: ~
railsrogerdev@railsdev:~$ curl 'http://localhost:3000'
200 OK
railsrogerdev@railsdev:~$ curl 'http://localhost:3000'
200 OK
railsrogerdev@railsdev:~$ curl -X POST 'http://localhost:3000'
405 Method Not Allowed
railsrogerdev@railsdev:~$ curl 'http://localhost:3000/kill'
curl: (52) Empty reply from server
railsrogerdev@railsdev:~$
```

Sending a POST request and KILL

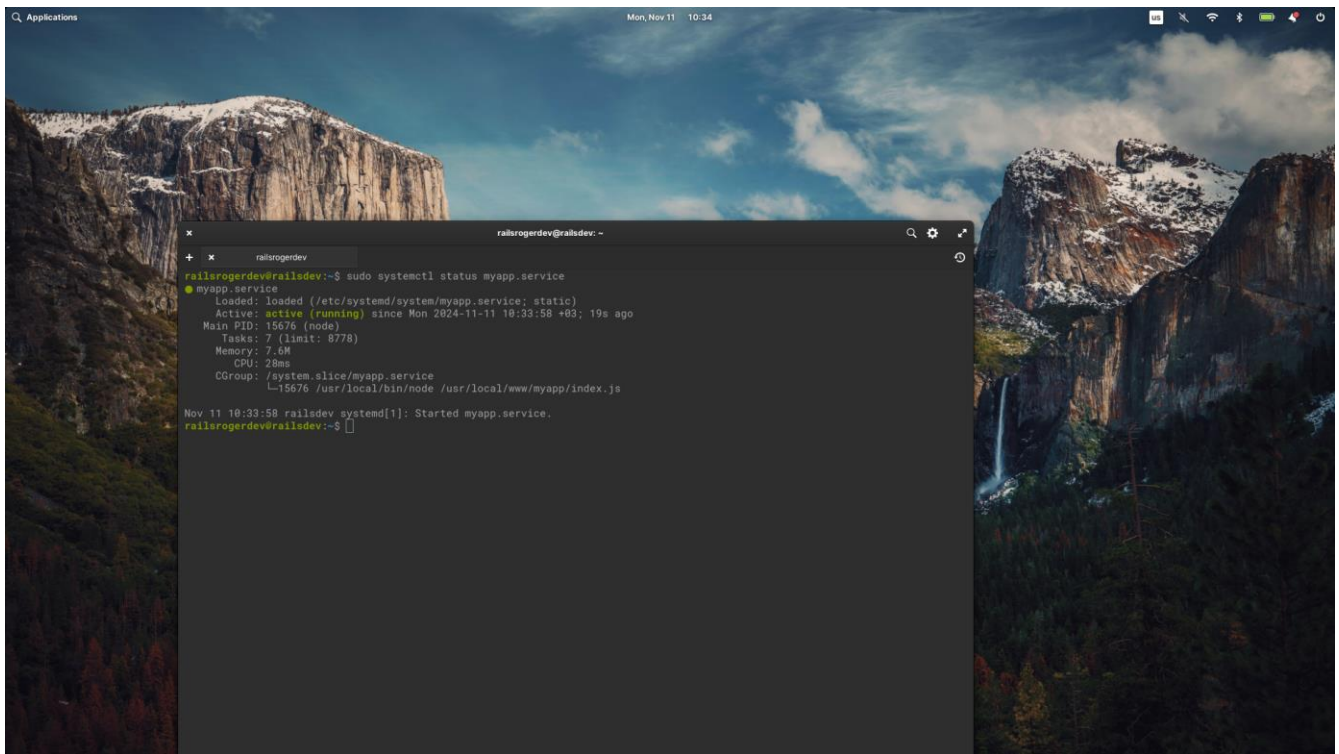
3. Creating a basic service.

I try to start the service:

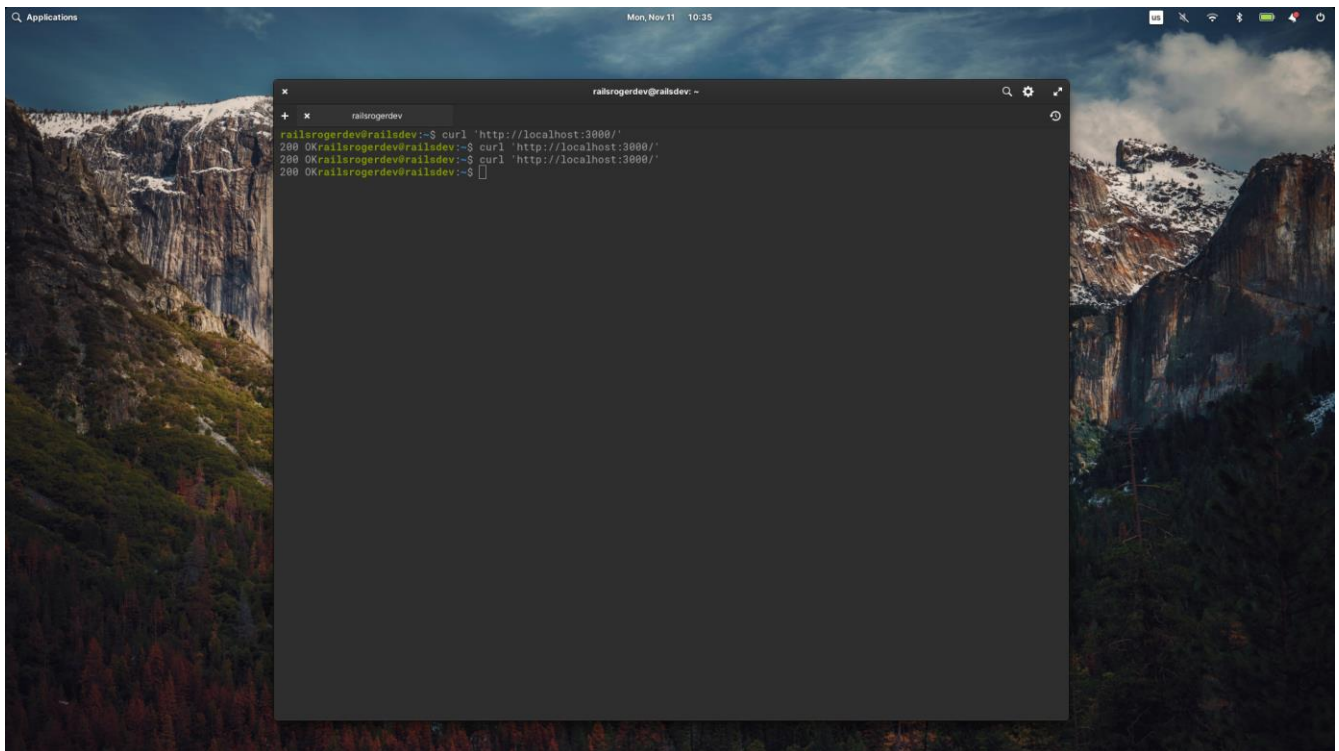
`sudo systemctl start myapp.service`

I check the status:

`sudo systemctl status myapp.service`

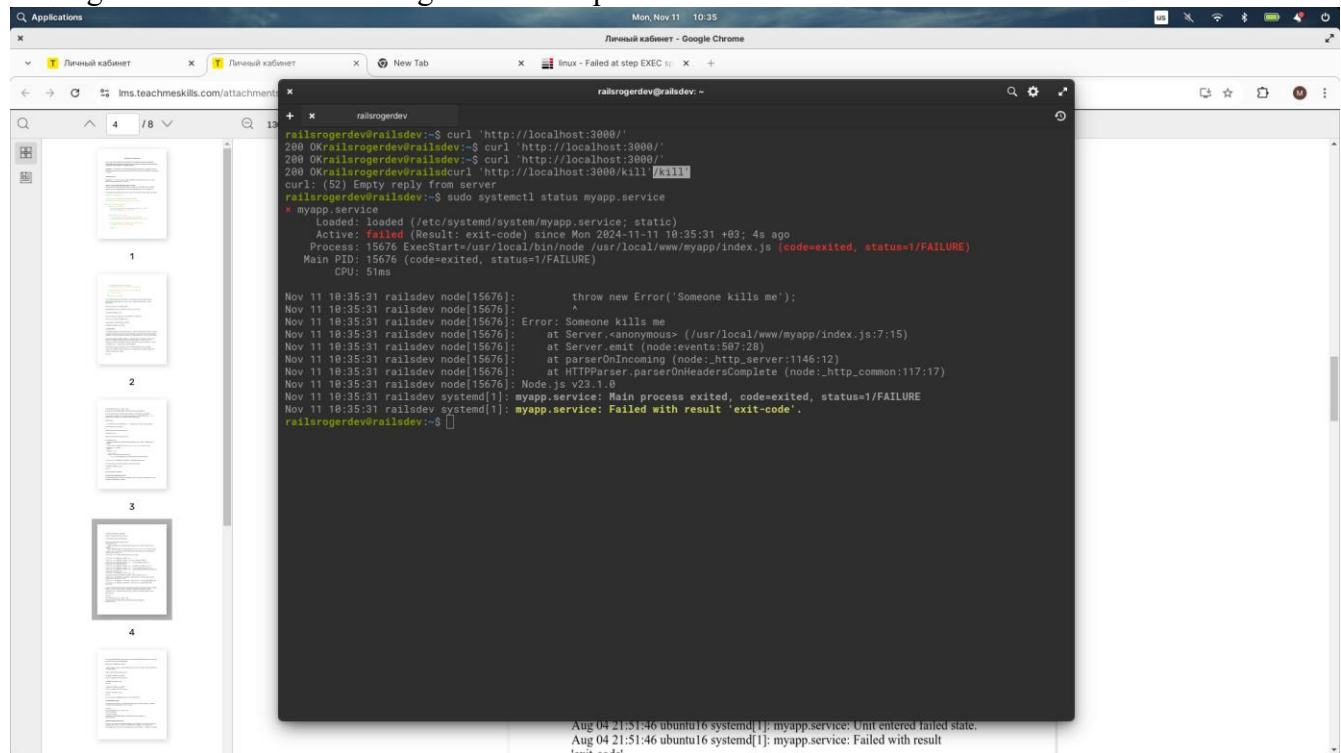


Test Service



Test Service 2

Testing the service before adding the restart option.

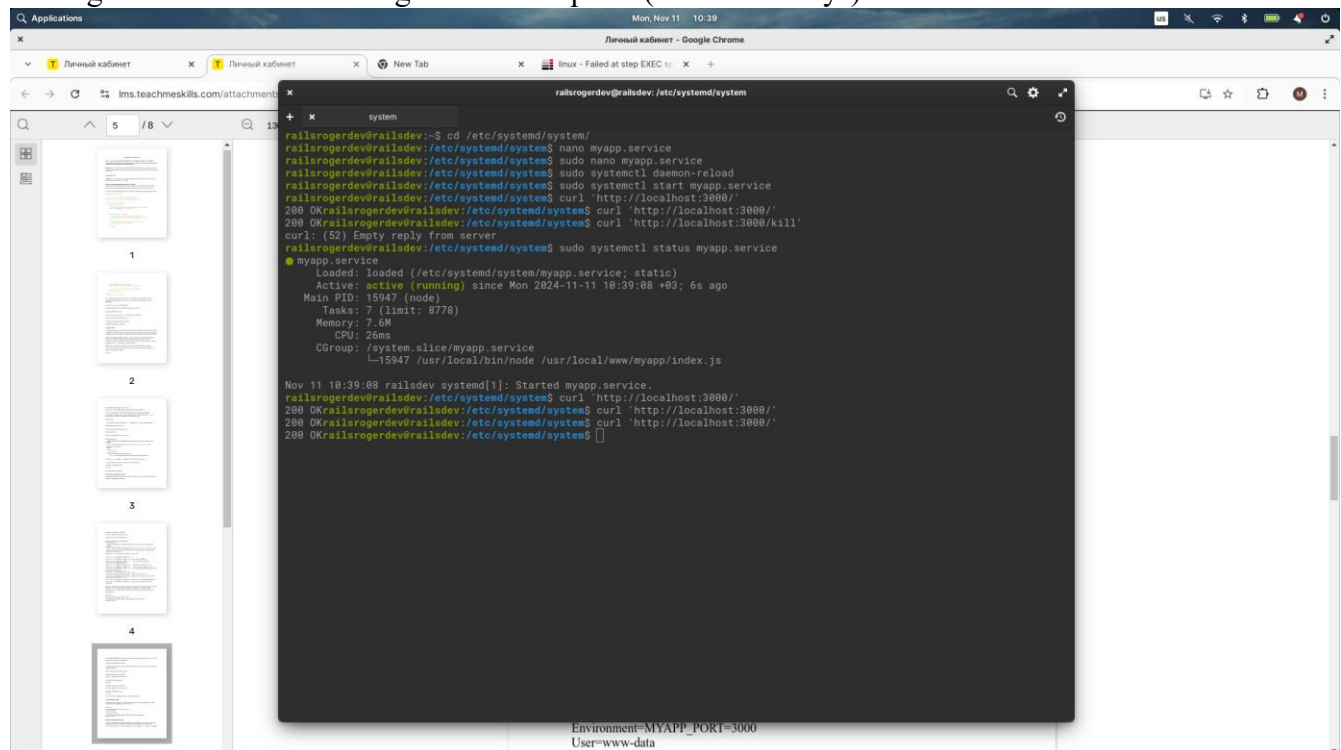


```
railsrogerdev@railsdev:~$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:~$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:~$ curl 'http://localhost:3000/kill'
curl: (52) Empty reply from server
railsrogerdev@railsdev:~$ sudo systemctl status myapp.service
● myapp.service
   Loaded: loaded (/etc/systemd/system/myapp.service; static)
   Active: failed (Result: exit-code) since Mon 2024-11-11 10:35:31 +03: 4s ago
     Process: 15676 ExecStart=/usr/local/bin/node /usr/local/www/myapp/index.js (code=exited, status=1/FAILURE)
    Main PID: 15676 (code=exited, status=1/FAILURE)
      CPU: 51ms

Nov 11 10:35:31 railsdev node[15676]:      throw new Error('Someone kills me');
Nov 11 10:35:31 railsdev node[15676]:      ^
Nov 11 10:35:31 railsdev node[15676]:      Error: Someone kills me
Nov 11 10:35:31 railsdev node[15676]:      at Server.<anonymous> (/usr/local/www/myapp/index.js:7:15)
Nov 11 10:35:31 railsdev node[15676]:      at Server.emit (node:events:507:20)
Nov 11 10:35:31 railsdev node[15676]:      at parserOnIncoming (node:_http_server:1146:12)
Nov 11 10:35:31 railsdev node[15676]:      at HTTPParser.parserOnHeadersComplete (node:_http_common:117:17)
Nov 11 10:35:31 railsdev node[15676]: Node.js v23.1.0
Nov 11 10:35:31 railsdev systemd[1]: myapp.service: Main process exited, code=exited, status=1/FAILURE
Nov 11 10:35:31 railsdev systemd[1]: myapp.service: Failed with result 'exit-code'.
railsrogerdev@railsdev:~$
```

Service Failed

Testing the service after adding the restart option (Restart=always)



```
railsrogerdev@railsdev:/etc/systemd/system$ cd /etc/systemd/system/
railsrogerdev@railsdev:/etc/systemd/system$ nano myapp.service
railsrogerdev@railsdev:/etc/systemd/system$ sudo nano myapp.service
railsrogerdev@railsdev:/etc/systemd/system$ sudo systemctl daemon-reload
railsrogerdev@railsdev:/etc/systemd/system$ sudo systemctl start myapp.service
railsrogerdev@railsdev:/etc/systemd/system$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:/etc/systemd/system$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:/etc/systemd/system$ curl 'http://localhost:3000/kill'
curl: (52) Empty reply from server
railsrogerdev@railsdev:/etc/systemd/system$ sudo systemctl status myapp.service
● myapp.service
   Loaded: loaded (/etc/systemd/system/myapp.service; static)
   Active: active (running) since Mon 2024-11-11 10:39:08 +03: 6s ago
     Main PID: 15947 (node)
       Tasks: 7 (limit: 8778)
      Memory: 7.0M
         CPU: 26ms
    CGroup: /system.slice/myapp.service
            └─15947 /usr/local/bin/node /usr/local/www/myapp/index.js

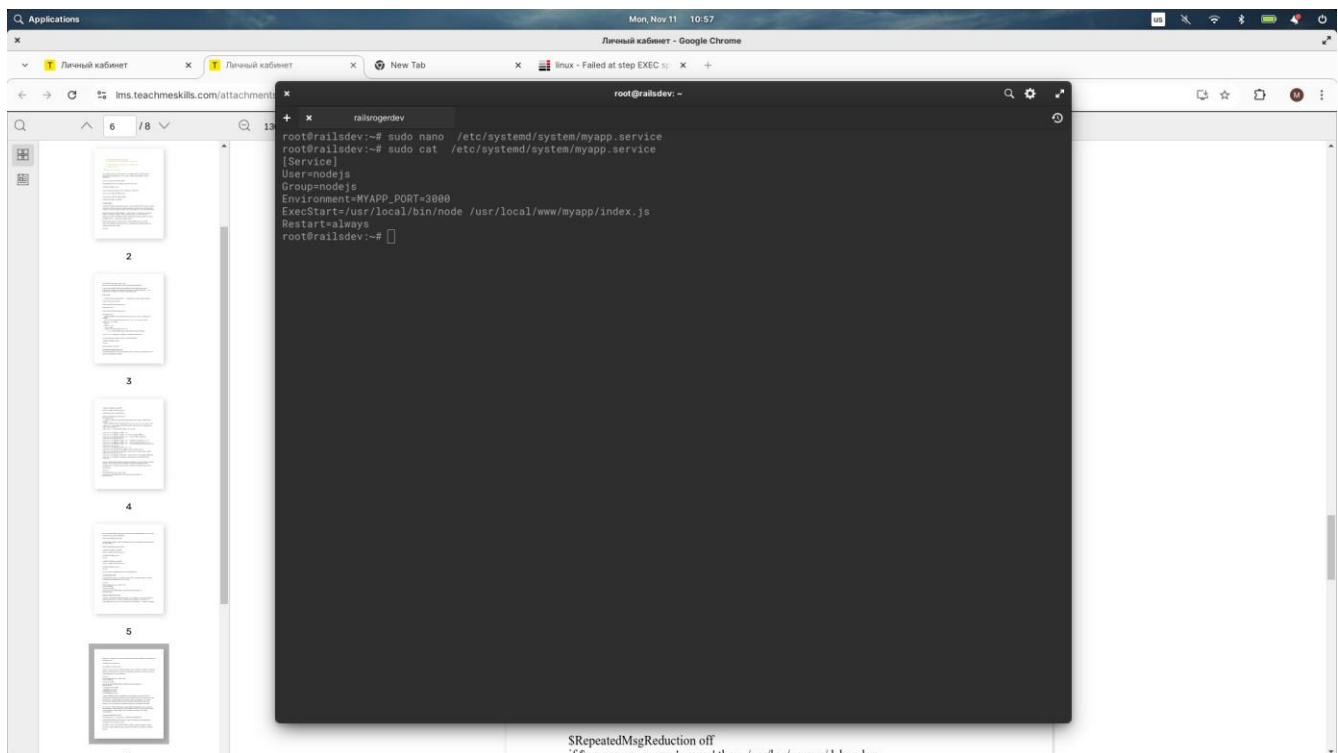
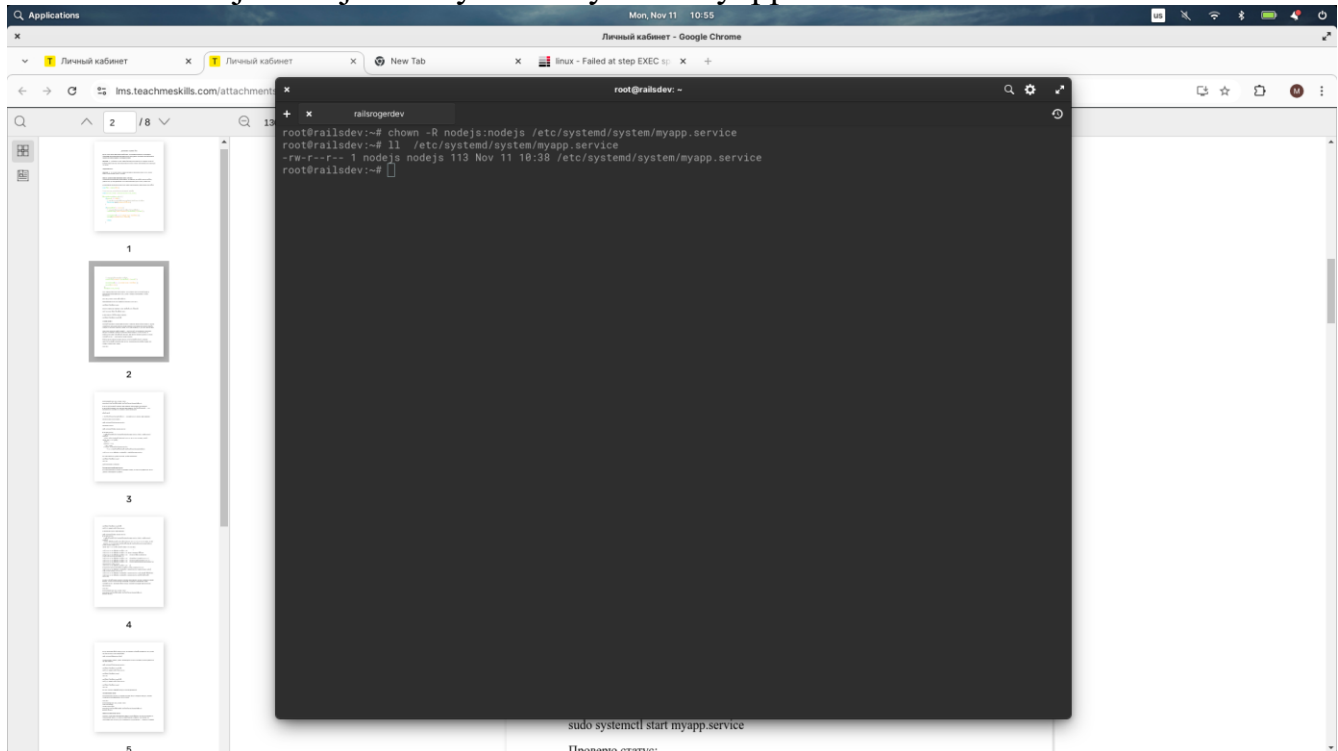
Nov 11 10:39:08 railsdev systemd[1]: Started myapp.service.
railsrogerdev@railsdev:/etc/systemd/system$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:/etc/systemd/system$ curl 'http://localhost:3000/'
200 OK
railsrogerdev@railsdev:/etc/systemd/system$
```

4. Adding a user and group, starting the service from a new user.

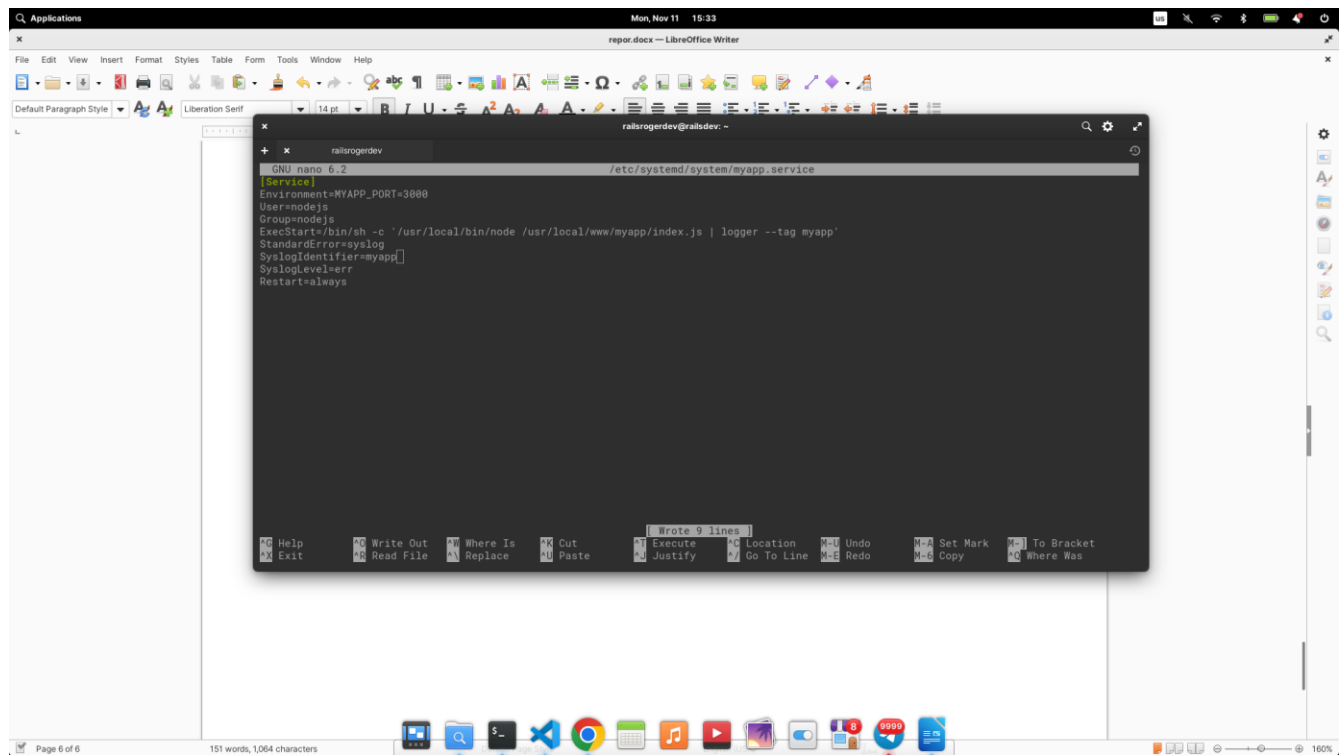
```
useradd -rs /bin/false nodejs
```

```
chown nodejs:nodejs /usr/local/bin/node
```

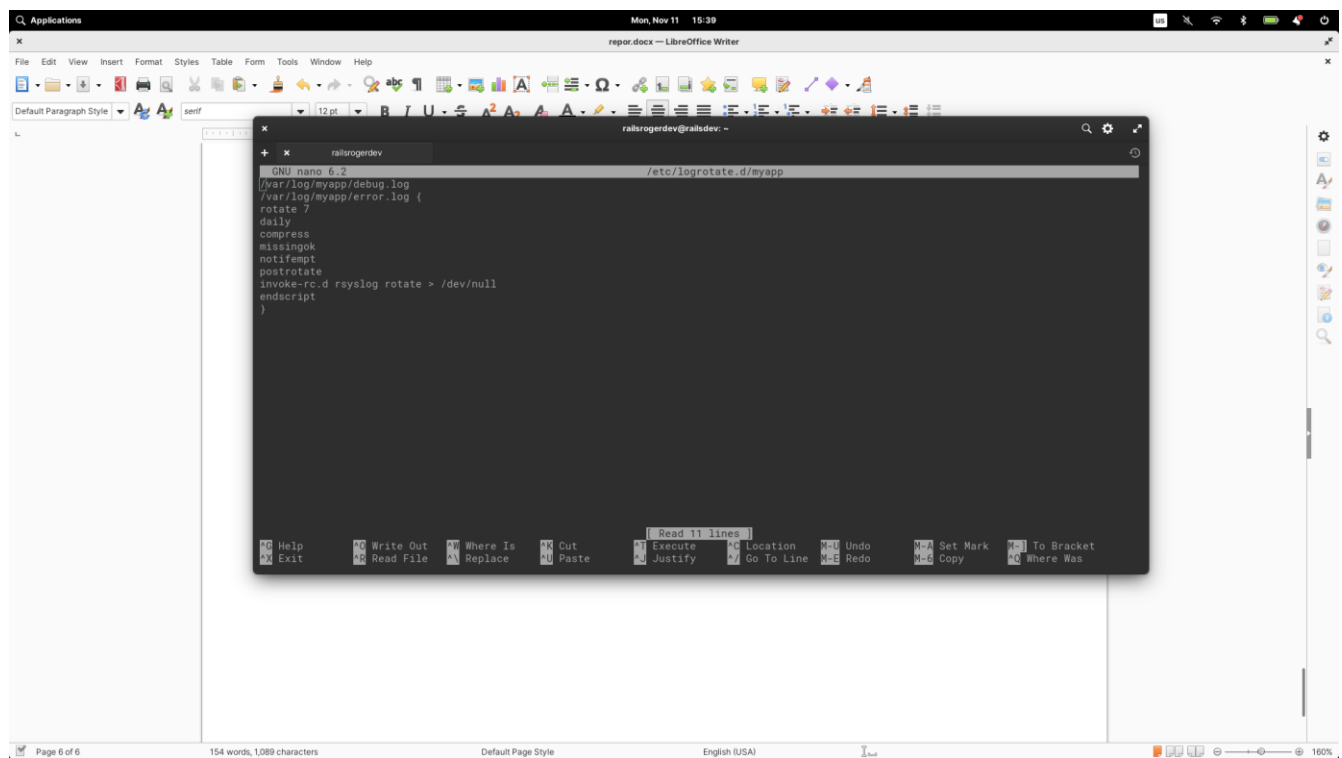
```
chown -R nodejs:nodejs /etc/systemd/system/myapp.service
```

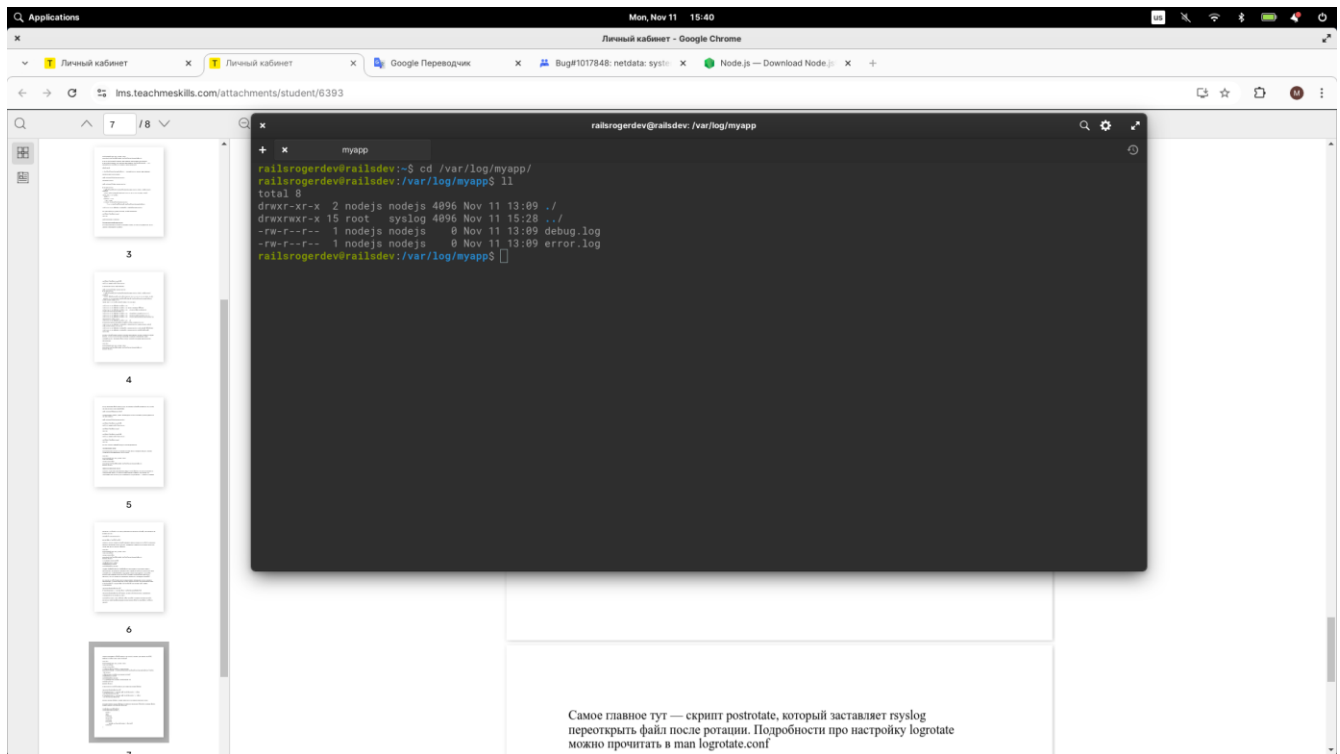


5. Recording and viewing logs.

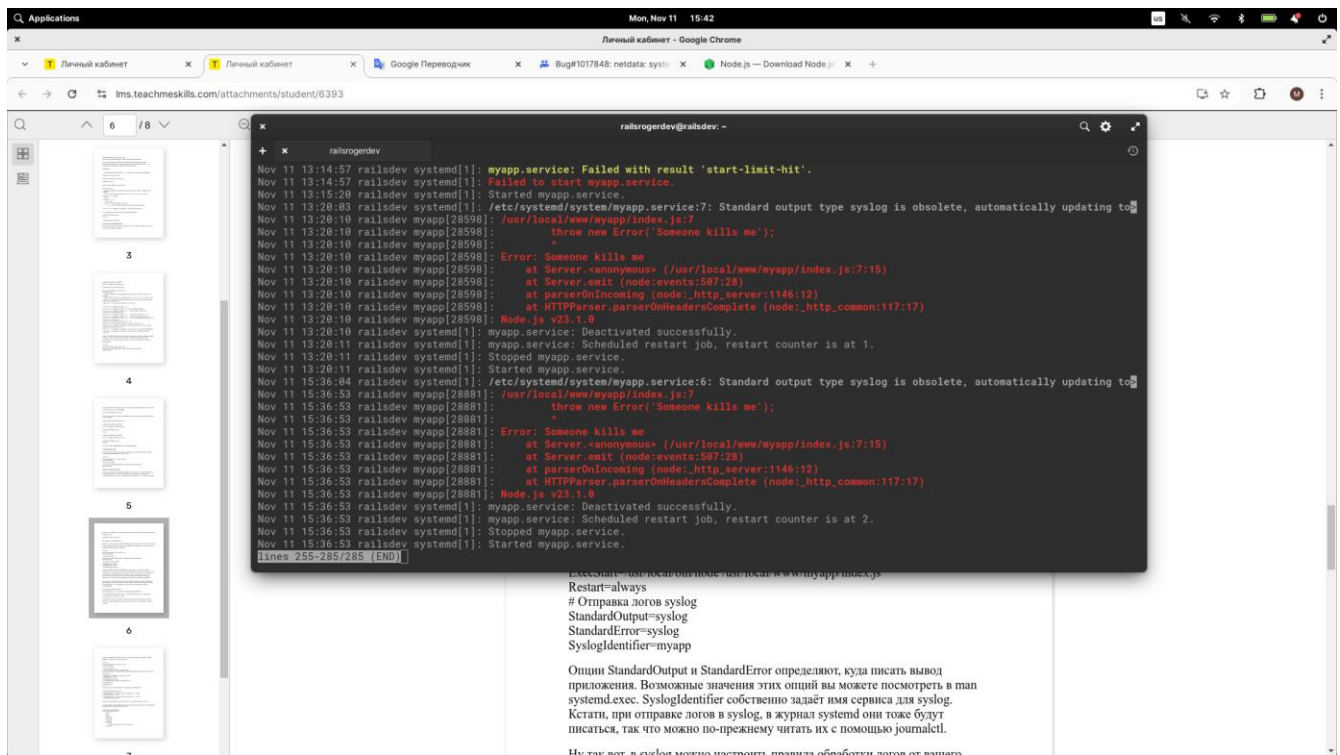


Modify file configuration





Granting access to the created user



Syslog