

Prometheus/ alertmanager / prometheus-bot

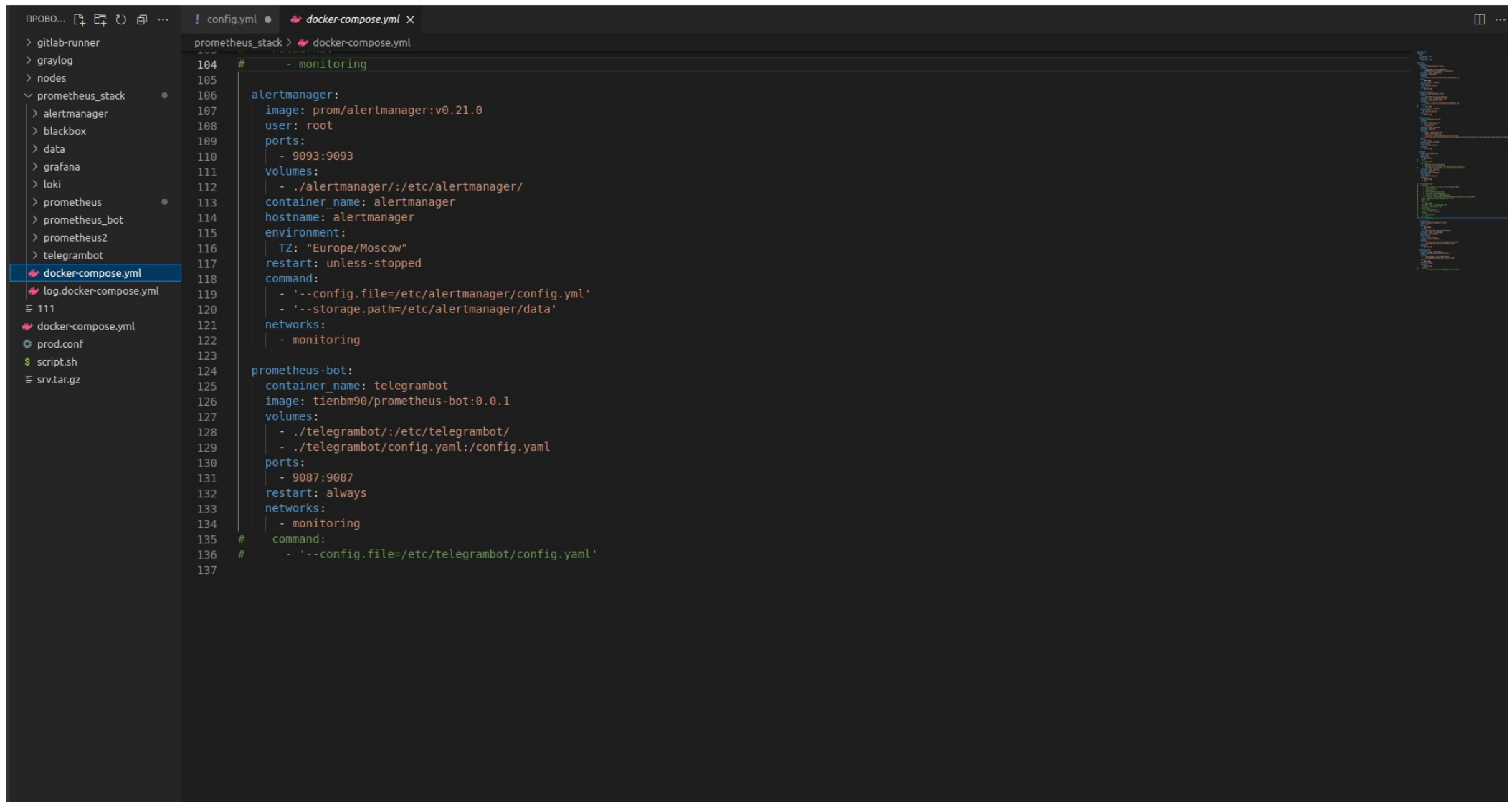
Дано:

- Развернут prometheus к нему подключены сервера с установленными на них node-exporter

Задание:

- Настроить alertmanager
- Создать телеграмм бота и настроить prometheus-bot

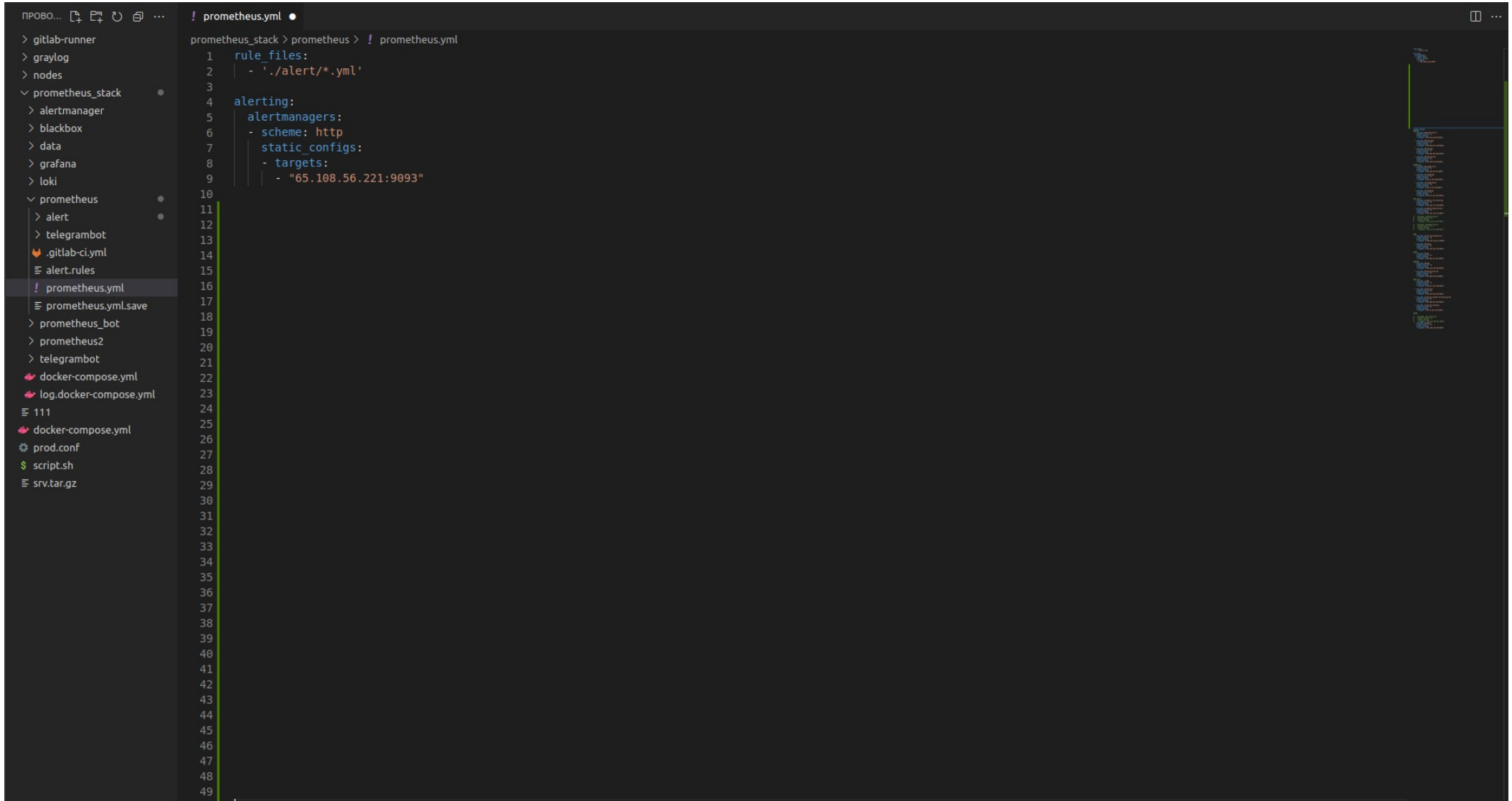
Добавляем alertmanager и prometheus-bot в docker-compose.yml



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like gitlab-runner, graylog, nodes, prometheus_stack, alertmanager, blackbox, data, grafana, loki, prometheus, prometheus_bot, prometheus2, telegrambot, docker-compose.yml, log.docker-compose.yml, prod.conf, script.sh, and srv.tar.gz. The code editor shows the docker-compose.yml file with the following content:

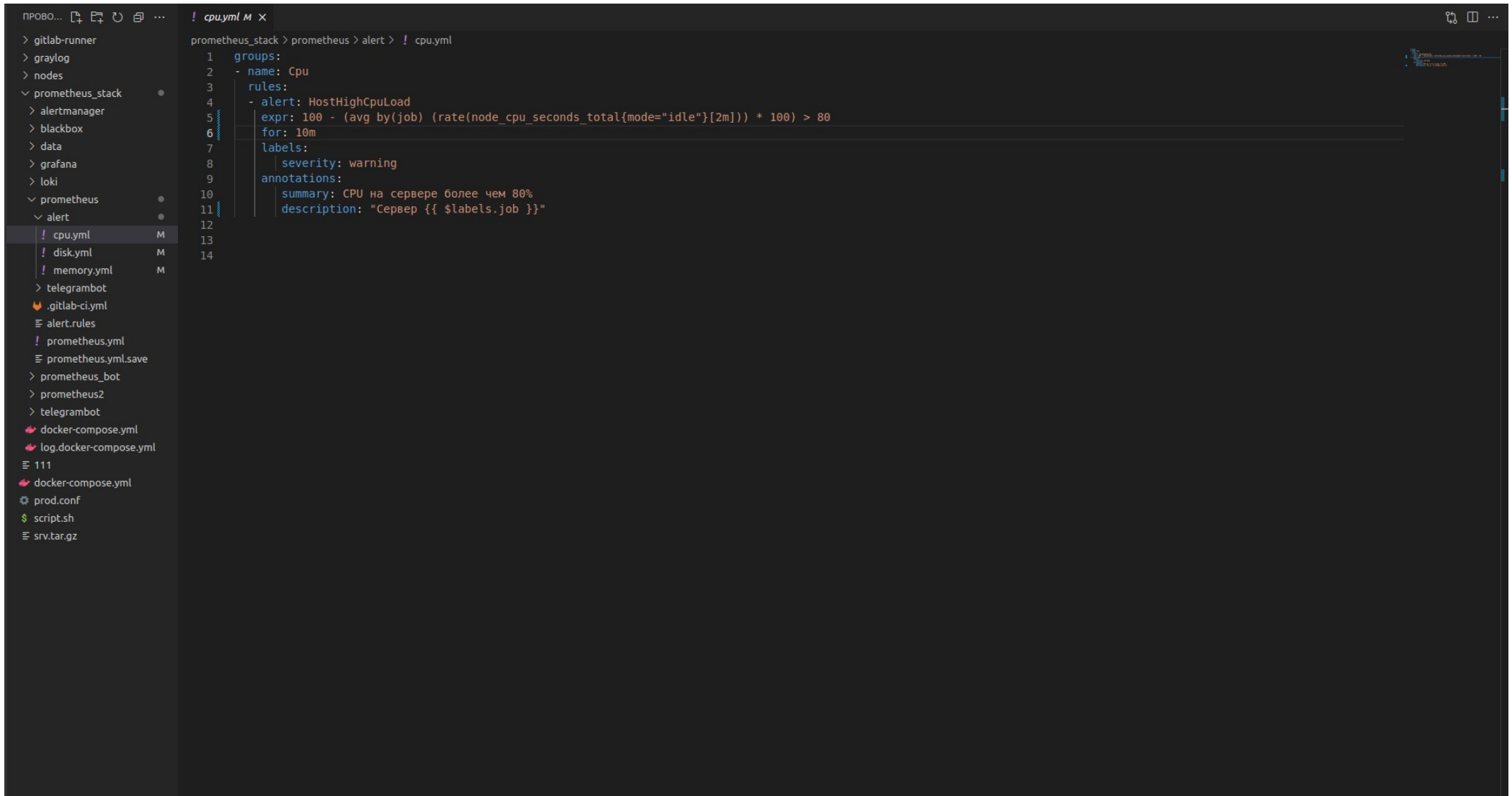
```
104 #
105     - monitoring
106
107 alertmanager:
108   image: prom/alertmanager:v0.21.0
109   user: root
110   ports:
111     - 9093:9093
112   volumes:
113     - ./alertmanager:/etc/alertmanager/
114   container_name: alertmanager
115   hostname: alertmanager
116   environment:
117     TZ: "Europe/Moscow"
118   restart: unless-stopped
119   command:
120     - '--config.file=/etc/alertmanager/config.yml'
121     - '--storage.path=/etc/alertmanager/data'
122   networks:
123     - monitoring
124
125 prometheus-bot:
126   container_name: telegrambot
127   image: tienbm90/prometheus-bot:0.0.1
128   volumes:
129     - ./telegrambot:/etc/telegrambot/
130     - ./telegrambot/config.yaml:/config.yaml
131   ports:
132     - 9087:9087
133   restart: always
134   networks:
135     - monitoring
136   command:
137     - '--config.file=/etc/telegrambot/config.yaml'
```

В конфигурацию prometheus прописывает alertmanager и папку где будут храниться файлы с алертами.



```
prometheus_stack > prometheus > ! prometheus.yml
1 rule_files:
2   - './alert/*.yaml'
3
4 alerting:
5   alertmanagers:
6     - scheme: http
7       static_configs:
8         - targets:
9           - "65.108.56.221:9093"
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
```

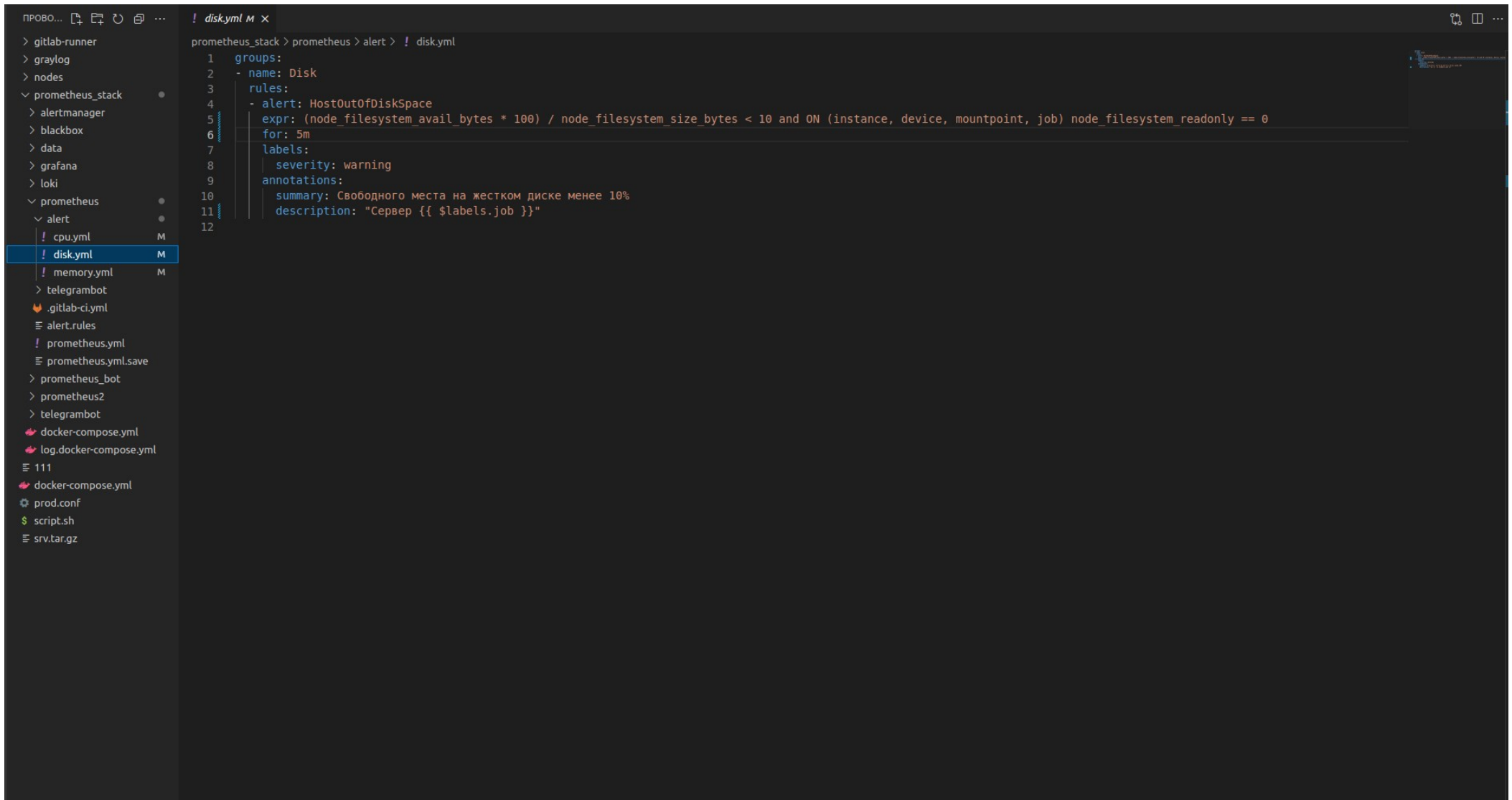
В файле `сри.yml` прописываем алерту которая сработает если цпу более 80% в течении 10 минут.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like `gitlab-runner`, `graylog`, `nodes`, `prometheus_stack`, `alertmanager`, `blackbox`, `data`, `grafana`, `loki`, `prometheus`, `alert`, `cpu.yml`, `disk.yml`, `memory.yml`, `telegrambot`, `.gitlab-ci.yml`, `alert.rules`, `prometheus.yml`, `prometheus.yml.save`, `prometheus_bot`, `prometheus2`, `telegrambot`, `docker-compose.yml`, `log.docker-compose.yml`, `111`, `docker-compose.yml`, `prod.conf`, `script.sh`, and `srv.tar.gz`. The code editor shows the content of `cpu.yml` with the following YAML configuration:

```
1 groups:
2   - name: Cpu
3     rules:
4       - alert: HostHighCpuLoad
5         expr: 100 - (avg by(job) (rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 80
6         for: 10m
7         labels:
8           severity: warning
9         annotations:
10          summary: CPU на сервере более чем 80%
11          description: "Сервер {{ $labels.job }}"
12
13
14
```

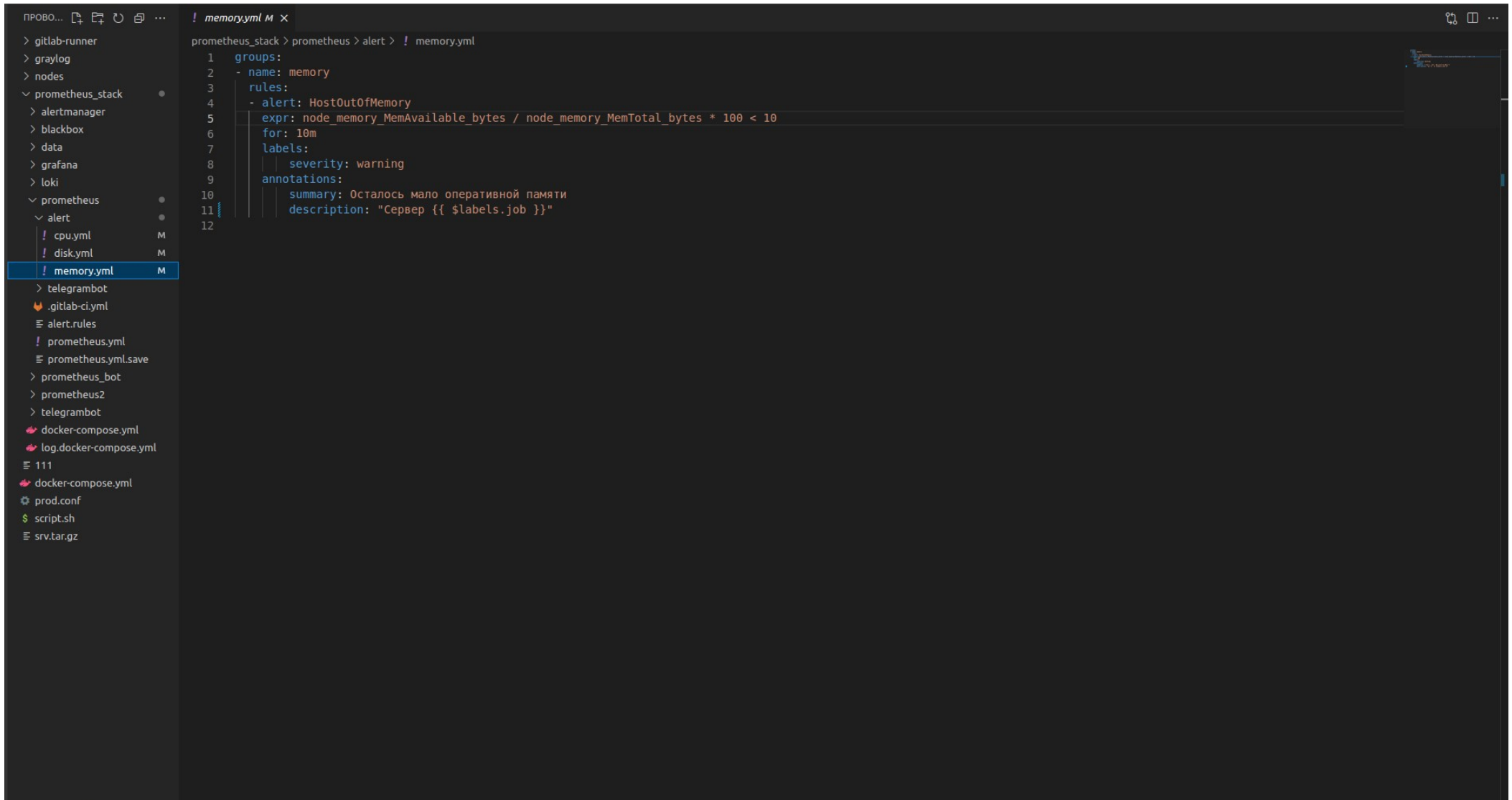
В файле disk.yml прописываем алерту которая сработает если свободного места на жестком диске менее 10%



```
prometheus_stack > prometheus > alert > ! disk.yml
1 groups:
2   - name: Disk
3     rules:
4       - alert: HostOutOfDiskSpace
5         expr: (node_filesystem_avail_bytes * 100) / node_filesystem_size_bytes < 10 and ON (instance, device, mountpoint, job) node_filesystem_readonly == 0
6         for: 5m
7         labels:
8           severity: warning
9         annotations:
10          summary: Свободного места на жестком диске менее 10%
11          description: "Сервер {{ $labels.job }}"
12
```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer lists various files and folders, including 'prometheus_stack', 'alertmanager', 'blackbox', 'data', 'grafana', 'loki', 'prometheus', 'alert', 'cpu.yml', 'disk.yml', 'memory.yml', 'telegrambot', '.gitlab-ci.yml', 'alert.rules', 'prometheus.yml', 'prometheus.yml.save', 'prometheus_bot', 'prometheus2', 'telegrambot', 'docker-compose.yml', 'log.docker-compose.yml', '111', 'docker-compose.yml', 'prod.conf', 'script.sh', and 'srv.tar.gz'. The 'disk.yml' file is selected and highlighted. The code editor shows the configuration for a Prometheus alert named 'HostOutOfDiskSpace'. The alert is defined within a 'groups' block under the name 'Disk'. The 'rules' block contains the alert definition, which includes an 'expr' (expression) that checks if the percentage of free space on the filesystem is less than 10% and if the filesystem is not read-only. The 'for' duration is set to 5m. The 'labels' block sets the severity to 'warning'. The 'annotations' block provides a summary and description for the alert.

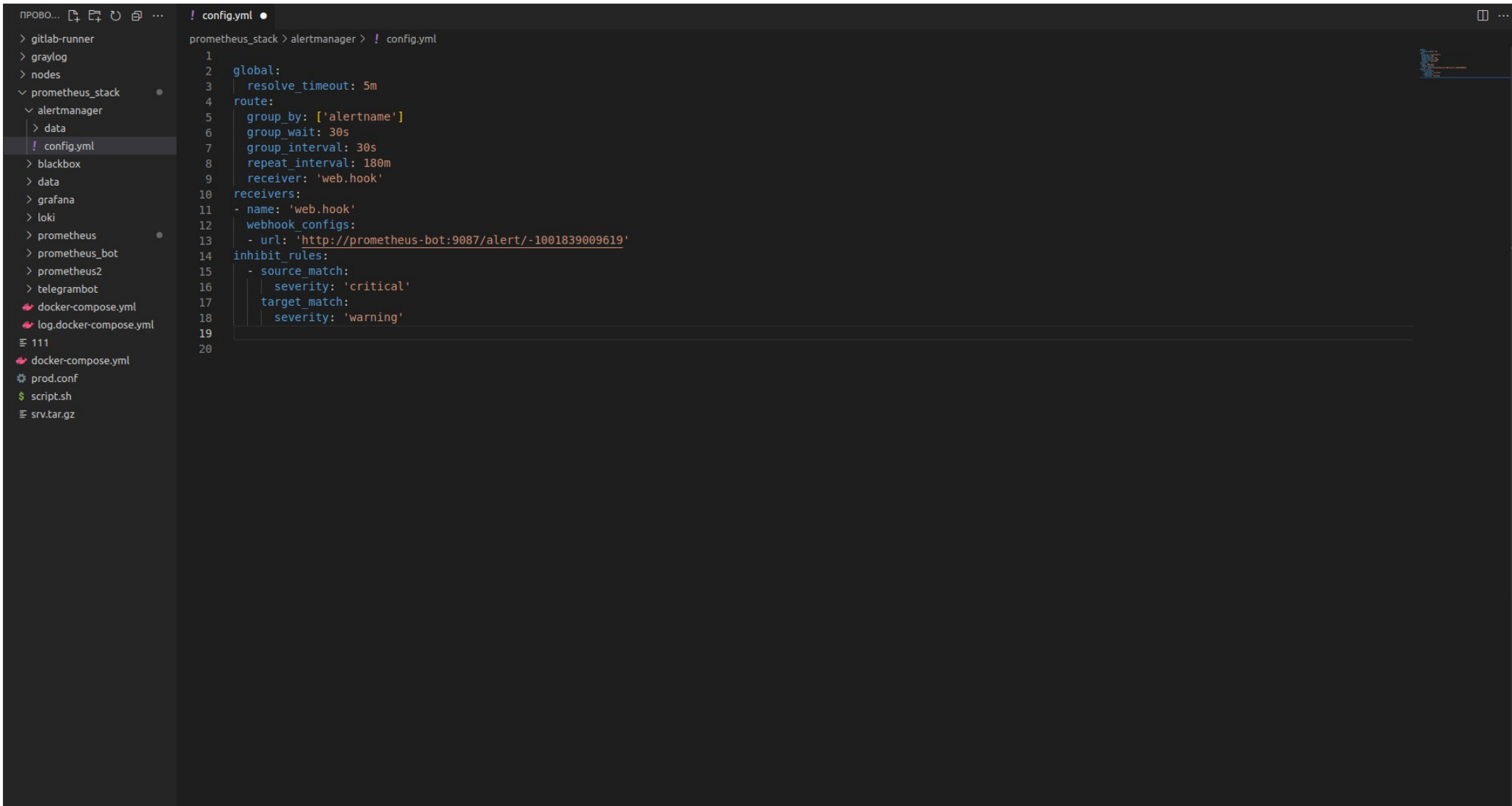
В файле memory.yml прописываем алерту которая сработает если оперативной памяти свободно менее 10%



```
prometheus_stack > prometheus > alert > ! memory.yml
1 groups:
2   - name: memory
3     rules:
4       - alert: HostOutOfMemory
5         expr: node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes * 100 < 10
6         for: 10m
7         labels:
8           severity: warning
9         annotations:
10          summary: Осталось мало оперативной памяти
11          description: "Сервер {{ $labels.job }}"
12
```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like `memory.yml`, `cpu.yml`, and `disk.yml`. The code editor shows the content of `memory.yml`, which defines a Prometheus alert rule named `HostOutOfMemory`. The rule triggers when the percentage of free memory is less than 10% for a duration of 10 minutes. The alert has a severity of `warning` and a summary in Russian: `Осталось мало оперативной памяти`. The description is `"Сервер {{ $labels.job }}"`.

Конфигурационный файл alertmanager



The image shows a code editor with a dark theme. On the left is a sidebar with a file explorer. The main area displays the content of a file named `! config.yml`. The file is a YAML configuration for Prometheus Alertmanager. The sidebar shows a tree structure with folders like `prometheus_stack` and `alertmanager`, and files like `! config.yml`, `blackbox`, `data`, `grafana`, `loki`, `prometheus`, `prometheus_bot`, `prometheus2`, `telegrambot`, `docker-compose.yml`, `log.docker-compose.yml`, `prod.conf`, `script.sh`, and `srv.tar.gz`. The main editor area shows the following YAML content:

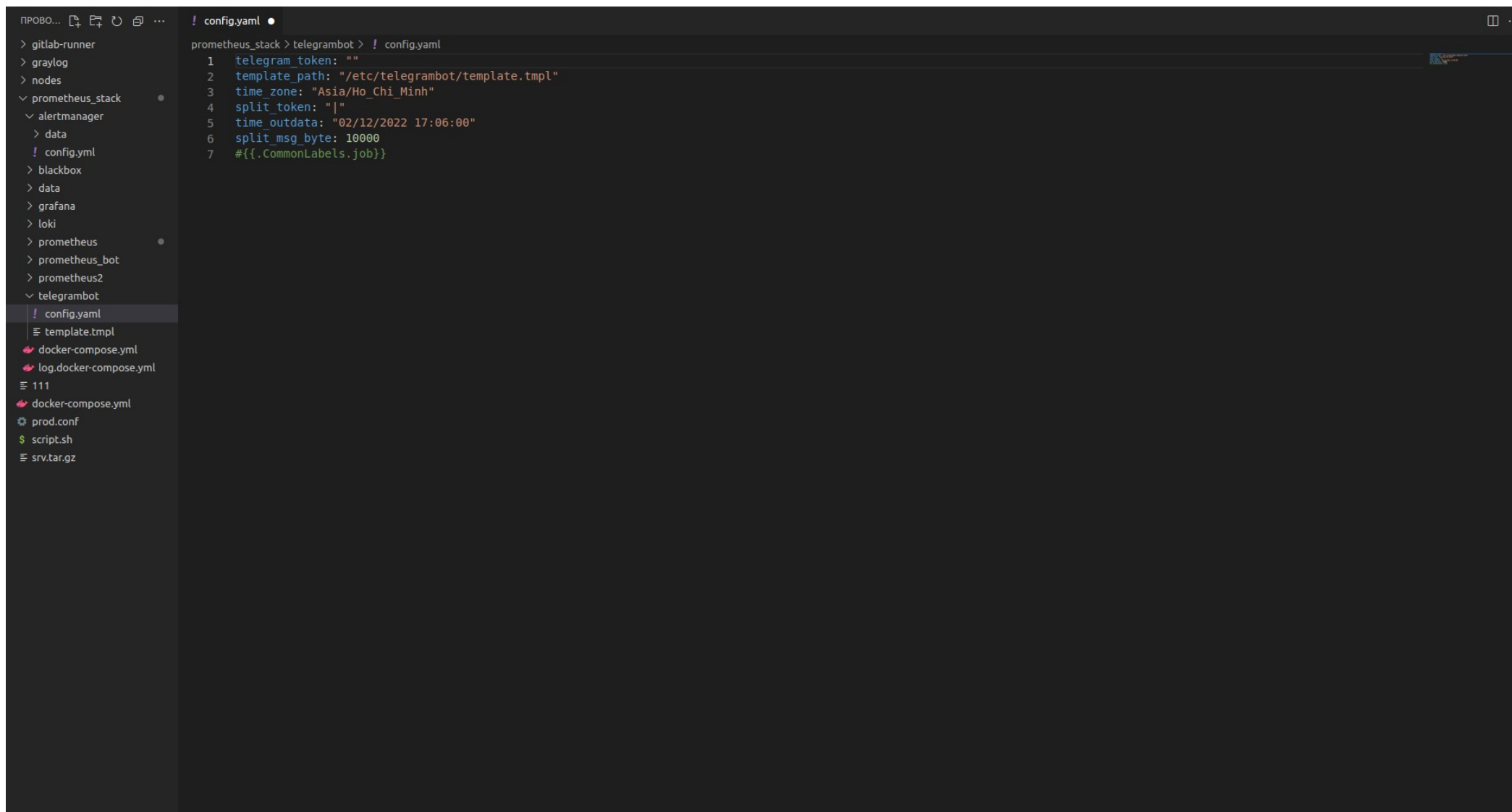
```
1
2 global:
3   resolve_timeout: 5m
4 route:
5   group_by: ['alertname']
6   group_wait: 30s
7   group_interval: 30s
8   repeat_interval: 180m
9   receiver: 'web.hook'
10 receivers:
11 - name: 'web.hook'
12   webhook_configs:
13   - url: 'http://prometheus-bot:9087/alert/-1001839009619'
14 inhibit_rules:
15 - source_match:
16   severity: 'critical'
17   target_match:
18   severity: 'warning'
19
20
```

Подключенные алерты в prometheus

The screenshot shows the Prometheus Alerts interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. On the right, there are settings, theme, and dark mode icons. Below the navigation bar, there's a status summary: 'Inactive (3)', 'Pending (0)', and 'Firing (0)'. A search bar with the placeholder 'Filter by name or labels' is also present. To the right of the search bar is a checkbox labeled 'Show annotations'. The main content area lists three alert rules, each with its file path and a status indicator:

- `/etc/prometheus/alert/cpu.yml > Cpu` (inactive)
 - > `HostHighCpuLoad` (0 active)
- `/etc/prometheus/alert/disk.yml > Disk` (inactive)
 - > `HostOutOfDiskSpace` (0 active)
- `/etc/prometheus/alert/memory.yml > memory` (inactive)
 - > `HostOutOfMemory` (0 active)

Конфигурационный файл prometheus-bot



The image shows a code editor with a dark theme. On the left is a file explorer sidebar showing a directory structure. The main editor area displays the content of a file named `config.yaml`.

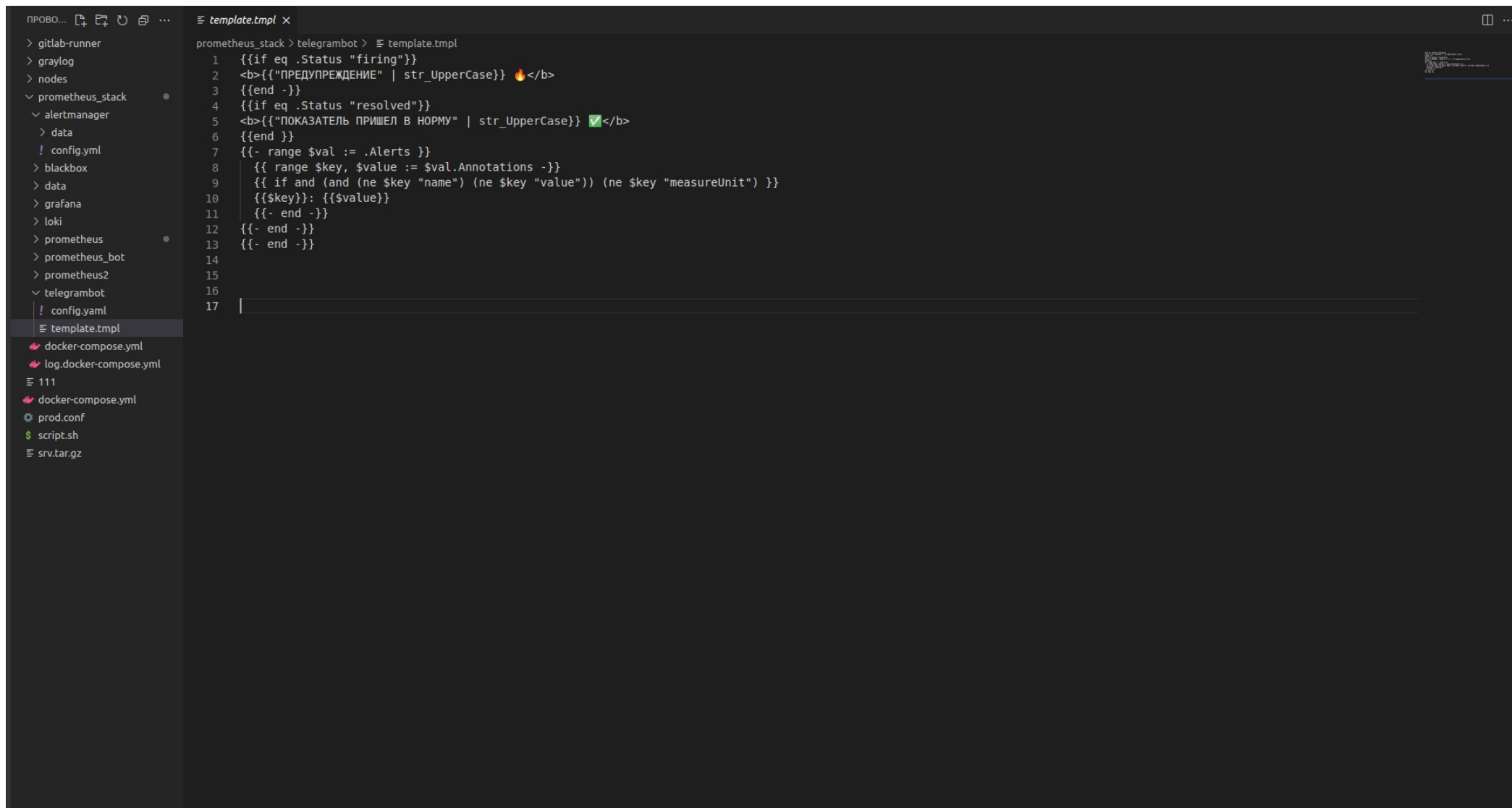
File Explorer (Left Sidebar):

- gitlab-runner
- graylog
- nodes
- prometheus_stack
 - alertmanager
 - data
 - config.yaml**
 - blackbox
 - data
 - grafana
 - loki
 - prometheus
 - prometheus_bot
 - prometheus2
 - telegrambot
 - config.yaml**
 - template.tmpl
- docker-compose.yml
- log.docker-compose.yml
- 111
- docker-compose.yml
- prod.conf
- script.sh
- srv.tar.gz

Editor Content (config.yaml):

```
prometheus_stack > telegrambot > ! config.yaml
1 telegram_token: ""
2 template_path: "/etc/telegrambot/template.tmpl"
3 time_zone: "Asia/Ho_Chi_Minh"
4 split_token: "|"
5 time_outdata: "02/12/2022 17:06:00"
6 split_msg_byte: 10000
7 #{{.CommonLabels.job}}
```

Шаблон сообщения в телеграм для prometheus-bot



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like `gitlab-runner`, `graylog`, `nodes`, `prometheus_stack`, `alertmanager`, `data`, `config.yml`, `blackbox`, `grafana`, `loki`, `prometheus`, `prometheus_bot`, `prometheus2`, `telegrambot`, `config.yml`, `template.tmpl`, `docker-compose.yml`, `log.docker-compose.yml`, `prod.conf`, `script.sh`, and `srv.tar.gz`. The code editor shows the content of `template.tmpl` with the following text:

```
prometheus_stack > telegrambot > template.tmpl
1  {{if eq .Status "firing"}}
2  <b>{{"ПРЕДУПРЕЖДЕНИЕ" | str_UpperCase}} 🔥</b>
3  {{end -}}
4  {{if eq .Status "resolved"}}
5  <b>{{"ПОКАЗАТЕЛЬ ПРИШЕЛ В НОРМУ" | str_UpperCase}} ✅</b>
6  {{end -}}
7  {{- range $val := .Alerts -}}
8  {{ range $key, $value := $val.Annotations -}}
9  {{ if and (and (ne $key "name") (ne $key "value")) (ne $key "measureUnit") }}
10  {{{key}}}: {{{value}}}
11  {{- end -}}
12  {{- end -}}
13  {{- end -}}
14
15
16
17
```

Создан телеграмм бот и проведен дебаг сообщений.

