

Rules и Alerts API

Alertmanagers

Endpoint: /api/v1/alertmanagers

Method: GET

Параметры запроса: -

Данный запрос позволяет получить список всех настроенных alertmanagers и их статус.

Пример:

```
curl 'http://localhost:9090/api/v1/alertmanagers'
```

Ответ должен быть примерно таким:

```
{
  "status":"success",
  "data":{
    "activeAlertmanagers":[
      {
        "url":"http://127.0.0.1:9093/api/v1/alerts"
      },
      {
        "url":"http://192.168.0.7:9093/api/v1/alerts"
      }
    ],
    "droppedAlertmanagers":[]
  }
}
```

Alerts

Endpoint: /api/v1/alerts

Method: GET

Параметры запроса: -

Данный запрос позволяет получить список активных алертов.

Пример:

```
curl http://localhost:9090/api/v1/alerts
```

Ответ должен быть примерно таким:

```
{
  "status": "success",
  "data": {
    "alerts": [

    ]
  }
}
```

Rules

Endpoint: /api/v1/rules

Method: GET

Параметры запроса: -

Данный запрос позволяет получить список всех правил: как record rules, так и alert rules.

Пример:

```
curl http://localhost:9090/api/v1/rules
```

Ответ должен быть примерно таким:

```
{
  "status": "success",
  "data": {
    "groups": [
      {
        "name": "slurm-edu-example-node-exporter-rules",
        "file": "/etc/prometheus/rules.yml",
        "rules": [
          {
            "name": "instance:node_cpus:count",
            "query": "count without(cpu, mode) (node_cpu_seconds_total{mode=\\\"idle\\\"})",
            "health": "ok",

```

```

    "type": "recording"
  },
  {
    "name": "instance_cpu:node_cpu_seconds_not_idle:rate5m",
    "query": "sum without(mode) (rate(node_cpu_seconds_total{mode!=\"idle\"}[5m]))",
    "health": "ok",
    "type": "recording"
  },
  {
    "name": "instance_mode:node_cpu_seconds:rate5m",
    "query": "sum without(cpu) (rate(node_cpu_seconds_total[5m]))",
    "health": "ok",
    "type": "recording"
  },
  {
    "name": "instance:node_cpu_utilization:ratio",
    "query": "sum without(mode)
(instance_mode:node_cpu_seconds:rate5m{mode!=\"idle\"}) / instance:node_cpus:count",
    "health": "ok",
    "type": "recording"
  }
],
"interval": 15
},
{
  "name": "NodeExporterGroup",
  "file": "/etc/prometheus/rules_alert.yml",
  "rules": [
    {
      "name": "ExporterDown",
      "query": "up == 0",
      "duration": 300,
      "labels": {
        "severity": "error"
      },
      "annotations": {
        "description": "Prometheus exporter down\n VALUE = {{ $value }}\n LABELS: {{
$labels }}"
      },
      "summary": "Exporter down (instance {{ $labels.instance }})"
    },
    {
      "alerts": [

```

```

        "health":"ok",
        "type":"alerting"
    },
    {
        "name":"HighCpuLoad",
        "query":"100 - (avg by(instance) (irate(node_cpu_seconds_total{mode=\"idle\"}[5m]))
* 100) \u003e 80",
        "duration":300,
        "labels":{
            "severity":"warning"
        },
        "annotations":{
            "description":"CPU load is \u003e 80%\n VALUE = {{ $value }}\n LABELS: {{
$labels }}",
            "summary":"High CPU load (instance {{ $labels.instance }})"
        },
        "alerts":[

    ],
        "health":"ok",
        "type":"alerting"
    },
    {
        "name":"SystemdServiceCrashed",
        "query":"node_systemd_unit_state{state=\"failed\"} == 1",
        "duration":300,
        "labels":{
            "severity":"warning"
        },
        "annotations":{
            "description":"SystemD service crashed\n VALUE = {{ $value }}\n LABELS: {{
$labels }}",
            "summary":"SystemD service crashed (instance {{ $labels.instance }})"
        },
        "alerts":[

    ],
        "health":"ok",
        "type":"alerting"
    }
    ],
    "interval":15
},

```

```
{
  "name": "PrometheusGroup",
  "file": "/etc/prometheus/rules_alert.yml",
  "rules": [
    {
      "name": "PrometheusConfigurationReload",
      "query": "prometheus_config_last_reload_successful != 1",
      "duration": 300,
      "labels": {
        "severity": "error"
      },
      "annotations": {
        "description": "Prometheus configuration reload error\n VALUE = {{ $value }}\n LABELS: {{ $labels }}",
        "summary": "Prometheus configuration reload (instance {{ $labels.instance }})"
      },
      "alerts": [
        ],
      "health": "ok",
      "type": "alerting"
    }
  ],
  "interval": 15
}
]
```