# **AlertManager Integrated to Prometheus**

Now, we need to configure the Prometheus server so it can talk to AlertManager service. We are going to set up an alert rule file which defines all rules needed to trigger an alert.

In the /etc/prometheus/prometheus.yml add the following

rule\_files:

- alert.rules.yml

alerting:

alertmanagers:

- static\_configs:

- targets:

- 'localhost:9093'

Which lead us to this final etc/prometheus/prometheus.yml file :

global:

scrape\_interval: 10s

rule\_files:

- alert.rules.yml

alerting:

alertmanagers:

- static\_configs:

- targets:

- 'localhost:9093'

scrape\_configs:

- job\_name: 'prometheus\_metrics'

scrape\_interval: 5s

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'node\_exporter\_metrics'

scrape\_interval: 5s

static\_configs:

- targets: ['localhost:9100','prometheus-target-1:9100','prometheus-target-2:9100']

Prometheus server is going to track incoming time series data, once any of the rules defined in etc/prometheus/alert.rules.yml is satisfied, an alert is triggered to AlertManager service that notifies the client on Slack.

nano /etc/prometheus/alert.rules.yml

groups:

- name: alert.rules

rules:

- alert: InstanceDown

expr: up == 0

for: 1m

labels:

severity: "critical"

annotations:

summary: "Endpoint {{ $labels.instance }} down"

description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 1 minutes."

The above alert rule checks whether the instance is down. Prometheus trigger an alert if it is down for more than 1 minute. We can check if the alert file is syntactically correct using “**promtool**” tool.

promtool check rules alert.rules.yml

Restart services :

sudo systemctl stop node\_exporter &&

sudo systemctl start node\_exporter &&

sudo systemctl stop prometheus &&

sudo systemctl start prometheus &&

sudo systemctl stop alertmanager &&

sudo systemctl start alertmanager

# 

# **—----------------------------------------------------------------------------------**

# **Prometheus Customized Alerts**

Here, we are going to define a set of rules in order to be alerted if the CPU load,Memory or Disk usage exceeds a certain threshold or if any instance of the supervised instances goes down.

Access the etc/prometheus/alert.rules.yml file and put the following :

|  | groups: |
| --- | --- |
|  | - name: alert.rules |
|  | rules: |
|  | - alert: InstanceDown |
|  | expr: up == 0 |
|  | for: 1m |
|  | labels: |
|  | severity: "critical" |
|  | annotations: |
|  | summary: "Endpoint {{ $labels.instance }} down" |
|  | description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 1 minutes." |
|  |  |
|  | - alert: HostOutOfMemory |
|  | expr: node\_memory\_MemAvailable / node\_memory\_MemTotal \* 100 < 25 |
|  | for: 5m |
|  | labels: |
|  | severity: warning |
|  | annotations: |
|  | summary: "Host out of memory (instance {{ $labels.instance }})" |
|  | description: "Node memory is filling up (< 25% left)\n VALUE = {{ $value }}\n LABELS: {{ $labels }}" |
|  |  |
|  |  |
|  | - alert: HostOutOfDiskSpace |
|  | expr: (node\_filesystem\_avail{mountpoint="/"} \* 100) / node\_filesystem\_size{mountpoint="/"} < 50 |
|  | for: 1s |
|  | labels: |
|  | severity: warning |
|  | annotations: |
|  | summary: "Host out of disk space (instance {{ $labels.instance }})" |
|  | description: "Disk is almost full (< 50% left)\n VALUE = {{ $value }}\n LABELS: {{ $labels }}" |
|  |  |
|  |  |
|  | - alert: HostHighCpuLoad |
|  | expr: (sum by (instance) (irate(node\_cpu{job="node\_exporter\_metrics",mode="idle"}[5m]))) > 80 |
|  | for: 5m |
|  | labels: |
|  | severity: warning |
|  | annotations: |
|  | summary: "Host high CPU load (instance {{ $labels.instance }})" |
|  | description: "CPU load is > 80%\n VALUE = {{ $value }}\n LABELS: {{ $labels }}" |

You can use these **queries** if you want to customize your **dashboard to get the CPU load , memory and disk usage.**

|  | Grafana Queries: |
| --- | --- |
|  | 1. CPU load : |
|  |  |
|  | sum(irate(node\_cpu\_seconds\_total{mode="idle",instance=~'$node'}[5m])) or sum(irate(node\_cpu{mode="idle",instance=~'$node'}[5m])) |
|  |  |
|  | 2. Memory Usage : |
|  |  |
|  | 100-(node\_memory\_MemAvailable{instance=~'$node'}/node\_memory\_MemTotal{instance=~'$node'}\*100) |
|  |  |
|  | 3. Disk Space Usage |
|  |  |
|  | 100-((node\_filesystem\_avail{instance=~'$node',mountpoint="/"} \* 100) / node\_filesystem\_size{instance=~'$node',mountpoint="/"}) |