Monitoring System setup

In the global configuration is possible to setup scare interval and evaluation interval.

global:

scrape\_interval: 15s

evaluation\_interval: 15s

Prometheus has to reference on Alert Manager, where messages will be published.

alerting:

alertmanagers:

- static\_configs:

- targets:

- localhost:9093

Every machine where Prometheus is installed can has its own alerting rules. In general alerting rules are located in the root folder of Prometheus.

rule\_files:

- "alert.rules"

- "node.rules"

- "test.rules"

Since there is a need to get more specific data, in N2Sky was decided to user Node Exporter Module. The reference on this module has to be added into configuration

- job\_name: 'node'

scrape\_interval: 5s

target\_groups:

* targets: ['localhost:9100']

Node Exporter Module has no configuration file. Prometheus listen the modules and scrap the data with a defined interval.

Deploying monitoring system:

For deploying alert manager Docker containers technology is used.

N2Sky Monitoring System API Documentation:

Alert Manager setup

All configuration of alert manager are written in YAML file.

On the beginning SMTP email sender should be configured. This would be used to sending notifications.

global:

smtp\_smarthost: 'localhost:25'

smtp\_from: 'alertmanager@example.org'

smtp\_auth\_username: 'alertmanager'

smtp\_auth\_password: 'password'

It is possible to define multiple Email templates and configure which template need to be loaded on which severe level. In configuration the path to templates need to be defined.

templates:

* '/etc/alertmanager/template/\*.tmpl'

When alerts are consumed they need to be converted using Email template and fired to the particular route. Every route has a receiver.

route:

group\_by: ['alertname', 'cluster', 'service']

group\_wait: 30s

group\_interval: 5m

repeat\_interval: 3h

receiver: team-X-mails

* group\_by. Group by label. This way ensures that multiple alerts from difference cluster can be received
* group\_wait. Ensures that multiple alerts can be fired shortly after particular group is received.
* group\_interval. Interval between alert batches.
* Receiver. Unique name of receiver which is defined in configuration.

Receiver it is a group of matching by regular expression events.

routes:

- match\_re:

service: ^(foo1|foo2|baz)$

receiver: team-X-mails

Receiver can be defined by user config. Reciever is an email wehere is alert notification will be send.

receivers:

- name: 'team-X-mails'

email\_configs:

- to: 'team-X+alerts@example.org'

Alerting rules:

The alerting rules are supporting simple query language, which looks very similar to Sequel Query Language.

There is multiple possibilities how work with a alerting rules. The query language allows to use an expression and as a result to check an attribute of time series.

ALERT HighLatency

IF api\_http\_request\_latencies\_second{quantile="0.7"} > 1

FOR 5m

LABELS { severity="critical"}

ANNOTATIONS { summary = "High latency detected ", description = "over limit” }

* All queries staring with “ALERT” namespace. After it follows name of alert in this case it is “HighLatency”.
* “IF” is a condition “api\_http\_request\_latencies\_second”, which based on Prometheus Tool expression. Set of time series with this expression has one parameter it is “quantile”. Reading condition as a whole can be translated in a human language like this: “Send a alert if latency request per second bigger then 0.7”.
* “FOR” it is period of time how often this condition should be checked.
* “LABELS” shows a severity level. There are 3 types of severity:
  + Critical
  + Warning
  + Page

Every severity level can be defined on developer needs.

* “ANNOTATIONS” shows a readable for human comments. There are two sub sections: summary, which shows a short description of the event and description where detailed information about deviation can be written.

Deploying alert manager:

For deploying alert manager Docker containers technology is used.

N2Sky Alerting System API Documentation: