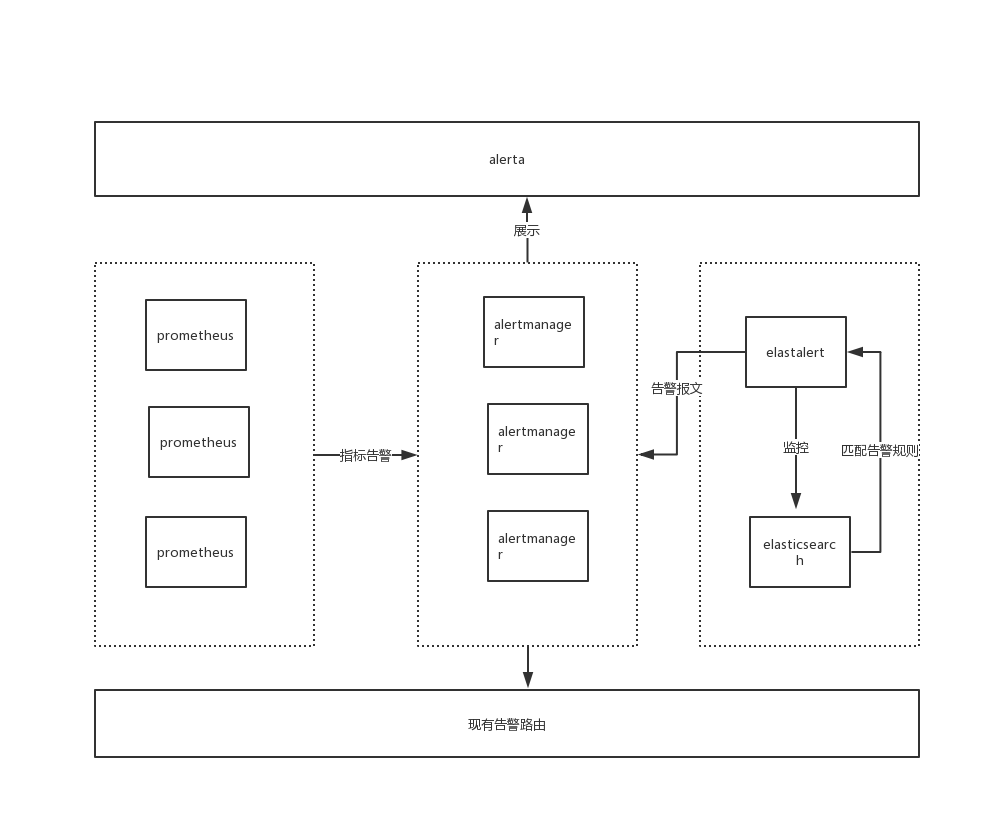
# 方案对比



# 系统架构



# 安装使用

## 3.1 安装配置

### 3.1.1 alerta

docker安装alerta服务命令如下

docker-compsoe -f alerta-compose.yaml up -d

alerta-compose.yaml文件内容：

version: '2.1'

services:

web:

image: alerta/alerta-web

ports:

- "8080:8080"

volumes:

- ./alerta-src/settings.py:/venv/lib/python3.6/site-packages/alerta/settings.py

- ./alerta-src/counts.html:/venv/lib/python3.6/site-packages/alerta//templates/oembed/counts.html

- ./alerta-src/oembed.py:/venv/lib/python3.6/site-packages/alerta//views/oembed.py

- ./alerta-src/alerta.py:/venv/lib/python3.6/site-packages/alerta/models/alarms/alerta.py

depends\_on:

- db

environment:

# - DEBUG=1 # remove this line to turn DEBUG off

- DATABASE\_URL=postgres://postgres:postgres@db:5432/monitoring

- AUTH\_REQUIRED=True

- ADMIN\_USERS=admin@alerta.io,devops@alerta.io #default password: alerta

- PLUGINS=sns,blackout,normalise,enhance

restart: always

db:

image: postgres

volumes:

- /data/pg-data:/var/lib/postgresql/data

environment:

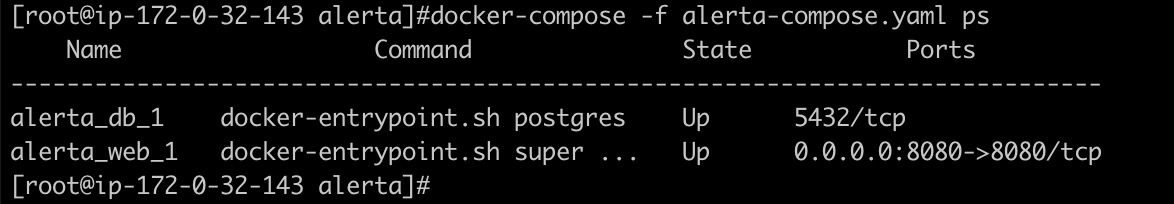
POSTGRES\_DB: monitoring

POSTGRES\_USER: postgres

POSTGRES\_PASSWORD: postgres

restart: always

alerta-src相关的文件主要涉及到severity相关的修改,安装成功后看到如下信息：



### 3.1.2 alertmanager推送信息至alerta配置

route:

receiver: devops

group\_by:

- alertname

routes:

- receiver: devops

continue: true

- receiver: alerta

repeat\_interval: 24h

group\_by: [“alertname”,”instance”,”timstamp”]

receivers:

- name: devops

webhook\_configs:

- send\_resolved: true

http\_config: {}

url: http://alertplus.apps.sit.caijj.net/alertplus/alert

- name: alerta

webhook\_configs:

- send\_resolved: true

http\_config:

basic\_auth:

username: admin@alerta.io

password: <secret>

url: http://172.0.32.143:8080/api/webhooks/prometheus

### 3.1.3 elastalert

docker 安装elastalert方式

git clone https://github.com/bitsensor/elastalert.git; cd elastalert

docker run -d -p 3030:3030 \

    -v `pwd`/config/elastalert.yaml:/opt/elastalert/config.yaml \

    -v `pwd`/config/config.json:/opt/elastalert-server/config/config.json \

    -v `pwd`/rules:/opt/elastalert/rules \

    -v `pwd`/rule\_templates:/opt/elastalert/rule\_templates \

    --net="host" \

--name elastalert bitsensor/elastalert:latest

kibana插件的安装

docker run -d --name kibana --net somenetwork -p 5601:5601 kibana:tag

./bin/kibana-plugin install plugin-相关版本

相关参考：<https://github.com/Yelp/elastalert>

<https://github.com/bitsensor/elastalert-kibana-plugin>

### 3.1.4 elastalert推送信息到alertmanager

如下标签，添加labels和annotations以及超时时间

alert:

- alertmanager

### alertmanager configuration ####

alertmanager\_host: http://172.0.32.143:9093

alertmanager\_alertname: "K8S Health Check Alert"

alertmanager\_labels: {"environment":"sit-elastalert","service":"K8S","instance":"{source}","severity":"warning","group":"k8s event health check"}

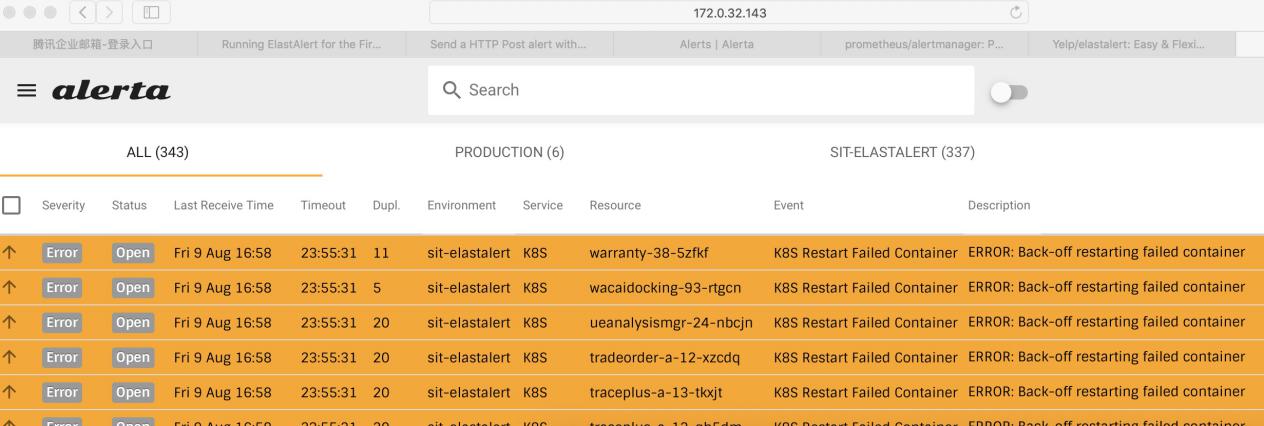
alertmanager\_annotations: {"description":"{message}"}

alertmanager\_timeout: 86400

## 3.2 使用说明

### 3.2.1 alerta

访问地址 <http://alerta-ip:8080>,默认用户名/密码 [admin@alert.io](mailto:admin@alert.io)/alerta



Duplicate：severity，environment，resource，event 相同

History: env，resource，event相同，severity或status不同

搜索框默认搜索description区域，其他区域搜索格式 label:value

基于environment标签页展示

可以根据时间范围，service，group，status过滤对应的告警

告警距收到最后一条告警默认24小时后超时，进入expired状态

告警收到resolve的消息进入closed状态

由housekeeping机制清理expired和closed状态的告警

Report:可以查看告警次数，状态变化等top排名

User ，group，role，keys界面提供功能较不完整，更多功能参考命令行alerta

prometheus标签对比alerta标签如下表:



### 3.2.2 elastalert

elastalert的主要操作就是配置rule，可以用plugin配置，也可以手工修改配置

下面以k8s监控日志健康检查告警为例说明:

es\_host: 172.0.13.13

es\_port: 9200

name: k8s health check

type: frequency

index: eventrouter\*

num\_events: 1

timestamp\_field: event\_lastTimestamp

timestamp\_type: iso

timeframe:

minutes: 5

filter:

- term:

event\_reason: Unhealthy

realert:

minutes: 0

alert:

- alertmanager

### alertmanager configuration ####

alertmanager\_host: http://172.0.32.143:9093

alertmanager\_alertname: "K8S Health Check Alert"

alertmanager\_labels: {"environment":"sit-elastalert","service":"K8S","instance":"{source}","severity":"warning","group":"k8s event health check"}

alertmanager\_annotations: {"description":"{message}"}

alertmanager\_timeout: 86400

3.2.2.1 any规则：匹配任何query查询到的记录都会触发alert

# From example\_rules/example\_frequency.yaml

# es里面包含get的记录都会匹配，时间片默认buffer time定义

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: any

index: nginx\*

filter:

- query:

query\_string:

query: get

realert:

minutes: 0

3.2.2.2 frequency规则：指定的timeframe内匹配的数量达到指定数

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: frequency

index: nginx\*

filter:

- query:

query\_string:

query: get

num\_events: 3

timeframe:

minutes: 5

realert:

minutes: 0

3.2.2.3 blacklist规则：某个field的值在列表中则匹配

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: blacklist

index: nginx\*

filter:

- query:

query\_string:

query: get

compare\_key: ResponseStatus

blacklist:

- k500

- k404

realert:

minutes: 0

3.2.2.4 whitelist规则：某个field的值不在列表中则匹配

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: whitelist

index: nginx\*

filter:

- query:

query\_string:

query: get

compare\_key: ResponseStatus

whitelist:

- k500

- k404

ignore\_null: true

realert:

minutes: 0

3.2.2.5 change规则：某field的值和上次不一致匹配告警

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: change

index: nginx\*

filter:

- query:

query\_string:

query: get

compare\_key: ResponseStatus

ignore\_null: true

timeframe:

minutes: 5

query\_key: type

realert:

minutes: 0

3.2.2.6 spike规则：当前timeframe中事件数是上一个timeframe事件数的多少倍上或下

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: spike

index: nginx\*

filter:

- query:

query\_string:

query: get

spike\_height: 3

spike\_type: up

timeframe:

minutes: 5

realert:

minutes: 0

3.2.2.7 percentage\_match规则：指定时间内满足事件的百分比触发告警

name: Example Percentage Match

type: percentage\_match

index: nginx\*

filter:

- term:

type: "get"

buffer\_time:

minutes: 5

match\_bucket\_filter:

- range:

ResponseStatus:

from: 501

to: 599

query\_key: id

max\_percentage: 50

realert:

minutes: 0

alert:

- debug

3.2.2.8 metric\_aggregation规则：指定时间内某field的值聚合满足最大或最小则匹配

name: Example Percentage Match

type: metric\_aggregation

index: nginx\*

filter:

- term:

type: "get"

buffer\_time:

minutes: 3

metric\_agg\_key: id

metric\_agg\_type: avg

max\_threshold: 4

realert:

minutes: 0

alert:

- debug

3.2.2.9 cardinality规则：指定时间内某字段出现的不同的值数目符合条件则匹配

# From example\_rules/example\_frequency.yaml

es\_host: 10.211.55.4

es\_port: 9200

name: Example rule

type: cardinality

index: nginx\*

filter:

- term:

type: get

cardinality\_field: ResponseStatus

timeframe:

minutes: 5

max\_cardinality: 3

realert:

minutes: 0