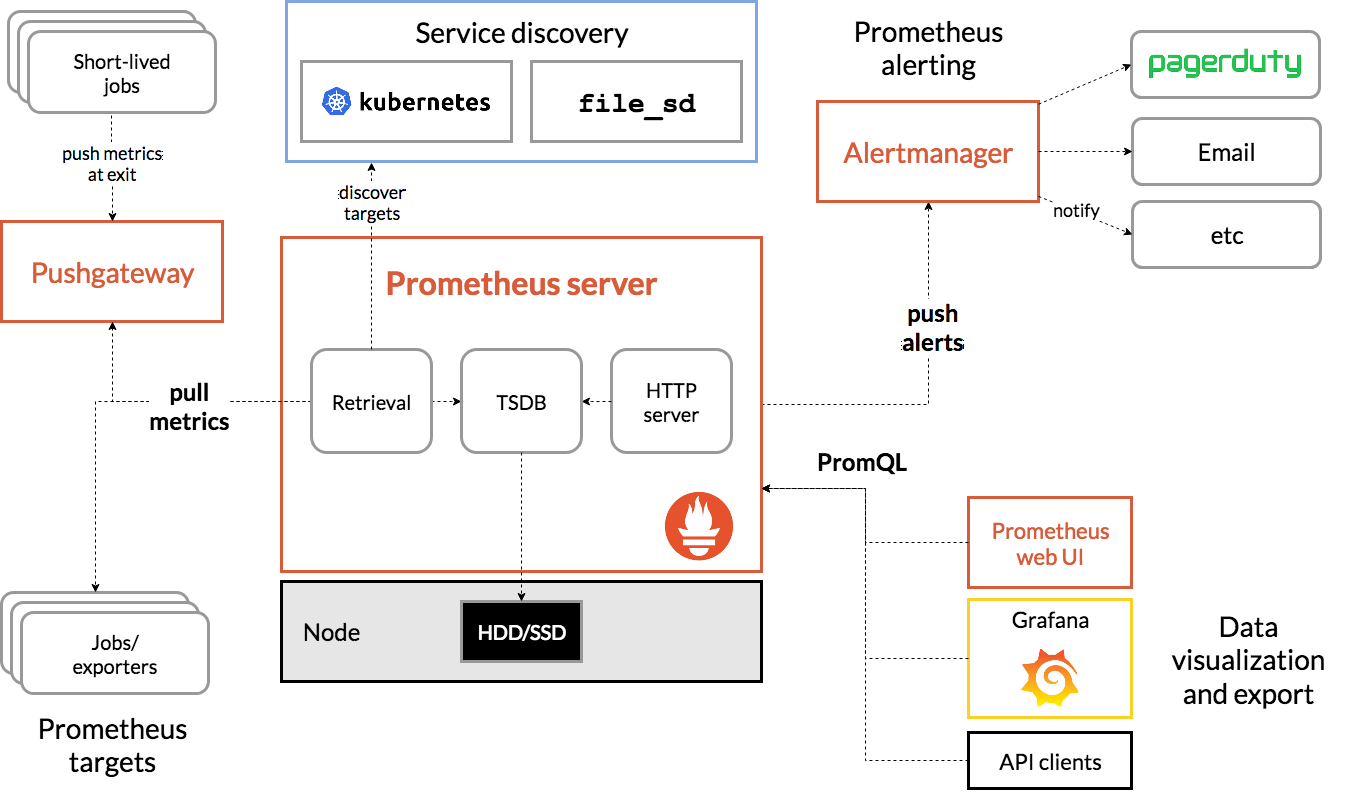
Ansible批量部署prometheus操作手册

中国移动（成都）产业研究院

2020年7月20日

#### Prometheus介绍

1、工作模型



2、适用范围

Prometheus非常适合用来获取和存储纯粹数值型时间序列数据，如cpu使用率、系统访问量、数据更新频率等，所以多被用来对宿主机和微服务架构中的指标监控。

Prometheus非常的可靠，每个Prometheus server可以作为一个独立体进行部署，不用依赖其他服务或者是网络。所以在底层基础设施出现问题时，你还可以从Prometheus server中取出历史指标来分析问题出现的原因，并且Prometheus server运行时也不会占用很多的资源

3、工作原理

Prometheus server利用各种各样的服务发现机制获取需要监控的target，Prometheus server通过pull从各种各样的target处拉取指标数据，数据可视化组件(Grfana)通过PromQl从Prometheus server查询数据，进行展示，Prometheus server根据自己定义的rule，可以提前对指标数据再次进行计算，触发报警的发送到alertmanager组件，alertmanager组件根据配置的告警方式发送相应的通知

4、组件介绍

1)、Prometheus server：主要的核心组件，用来收集和存储时间序列数据

2)、client libraries：提供个客户端，主要是用来帮助应用程序更容易生成满足Prometheus格式的监控数据，支持各种各样的开发语言

3)、push gateway：对于那些生存时间很短的job工作，采用Prometheus的pull模式可能来不及收集，可以部署这个组件，让job主动把监控指标push到getway，Prometheus再从getway中拉取

4)、各种各样的exports

5)、Alertmanager：一个告警组件

#### 部署步骤说明

1、ansible管理机完成部署prometheus部分exporter，本手册以node\_exporter、nginx\_exporter和mysqld\_exporter为例

2、通过ansible工具批量部署node\_exporter、nginx\_exporter、mysqld\_exporter到所有主机，其他监控服务类似。

#### 配置Yum源

1、阿里云YUM源文件下载

cd /etc/yum.repos.d/

wget http://mirrors.aliyun.com/repo/Centos-7.repo

2、阿里云YUM源缓存

yum clean all && yum repolist && yum makecache

#### 下载安装包及Exporter

mkdir -p /usr1/software

cd /usr1/software

wget <https://github.com/prometheus/prometheus/releases/download/v2.8.1/prometheus-2.8.1.linux-amd64.tar.gz>

# prometheus安装包

wget https://dl.grafana.com/oss/release/grafana-7.0.6.linux-amd64.tar.gz

#grafana安装包

wget <https://github.com/prometheus/alertmanager/releases/download/v0.15.2/alertmanager-0.15.2.linux-amd64.tar.gz>

#alertmanager安装包

wget <https://prometheus.io/download/#pushgateway>/[pushgateway-1.2.0.linux-amd64.tar.gz](https://github.com/prometheus/pushgateway/releases/download/v1.2.0/pushgateway-1.2.0.linux-amd64.tar.gz)

#pushgateway安装包

wget <https://grafana.com/api/dashboards/8919/revisions/17/download>

#node\_exporter下载

wget <https://github.com/prometheus/mysqld_exporter/releases/download/v0.10.0/mysqld_exporter-0.10.0.linux-amd64.tar.gz>

# mysqld\_exporter下载

wget <http://download.baiyongjie.com/linux/prometheus/nginx-vts-exporter-0.10.3.linux-amd64.tar.gz>

#nginx\_exporter下载

wget https://github.com/prometheus/pushgateway/releases/download/v1.2.0/pushgateway-1.2.0.linux-amd64.tar.gz

#pushgateway下载

wget

https://github.com/prometheus/blackbox\_exporter/releases/download/v0.17.0/blackbox\_exporter-0.17.0.linux-amd64.tar.gz

#blackbox\_exporter下载

#### 部署基础服务

##### 一、Prometheus\_server部署

tar -zxvf prometheus-2.8.1.linux-amd64.tar.gz -C /usr/local/

cd /usr/local

mv prometheus-2.8.1.linux-amd64/ prometheus

cd prometheus/

 ./prometheus –version

修改prometheus.yml文件，确定启动ip

vim /usr/local/prometheus/prometheus.yml

- job\_name: 'prometheus'

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

scrape\_interval: 5s

static\_configs:

- targets: ['192.168.12.129:9090']

labels:

instance: prometheus

完成修改后，可以直接启动

./prometheus

添加用户，后期用此账号启动服务

groupadd prometheus

useradd -g prometheus -s /sbin/nologin prometheus

赋权和创建prometheus运行数据目录

chown -R prometheus:prometheus /usr/local/prometheus/

mkdir -p /xia/software/prometheus-data

chown -R prometheus:prometheus /xia/software/prometheus-data

再次修改配置文件(添加剩余主机)

- job\_name: 'node'

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

scrape\_interval: 10s

static\_configs:

- targets: ['192.168.12.129:9100']

labels:

instance: master

- targets: ['192.168.12.130:9100']

labels:

instance: node1

- targets: ['192.168.12.131:9100']

labels:

instance: node2

启动并验证

1. 启动

**/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" --storage.tsdb.path=/xia/software/prometheus-data &**

1. 查看服务

netstat -antup | grep prometheus

1. web ui

<http://192.168.12.129:9090/targets>

说明：Statu-->Targets展示监控具体的监控目标

centos还没有监控到，一会安装插件，目前State为DOWN

在Status菜单下，Configuration，Rule，Targets等，

Statu-->Configuration展示prometheus.yml的配置，如下

http://192.168.12.129:9090/config

<http://192.168.12.129:9090/rules>

<http://192.168.12.129:9090/graph>

http://192.168.12.129:9090/alerts

1. 绘图

访问http://192.168.33.129:9090/metrics  查看从exporter具体能抓到的数据

##### 二、Node\_exporter部署

Node\_exporter收集机器的系统数据，这里采用prometheus官方提供的exporter，除node\_exporter外，官方还提供consul，memcached，haproxy，mysqld等exporter，具体可查看官网

这里在prometheus服务和 node节点部署相关服务

tar -zxvf node\_exporter-1.0.1.linux-amd64.tar.gz -C /usr/local/

cd /usr/local/

mv node\_exporter-1.0.1.linux-amd64 node\_exporter

设置用户（节点）

groupadd prometheus

useradd -g prometheus -s /sbin/nologin prometheus

chown -R prometheus:prometheus /usr/local/node\_exporter

起服务

**/usr/local/node\_exporter/node\_exporter &**

查看服务

netstat -antup | grep node\_exporter

Statu-->Targets展示监控具体的监控目标

修改prometheus.yml文件，确定启动ip

- job\_name: 'node'

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

scrape\_interval: 10s

static\_configs:

- targets: ['192.168.12.129:9100']

labels:

instance: master

- targets: ['192.168.12.130:9100']

labels:

instance: node1

- targets: ['192.168.12.131:9100']

labels:

instance: node2

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

<http://192.168.12.129:9090/targets>

grafana展示dashboard手动编辑导入

wget https://grafana.com/api/dashboards/405/revisions/1/download

手动导入

##### 三、Mysqld\_exporter部署

tar zxvf mysqld\_exporter-0.10.0.linux-amd64.tar.gz -C /usr/local/

mv mysqld\_exporter-0.10.0.linux-amd64 mysqld\_exporter

mysql -uroot -p123456

mysql> GRANT REPLICATION CLIENT,PROCES ON \*.\* TO 'mysql\_monitor'@'localhost' identified by 'mysql\_monitor';

mysql> GRANT ALL ON \*.\* TO 'mysql\_monitor'@'localhost';

创建my.cnf配置文件

vim /usr/local/mysqld\_exporter/my.cnf

[client]

user=mysql\_monitor

password=mysql\_monitor

运行服务

**/usr/local/mysqld\_exporter/mysqld\_exporter --config.my-cnf="/usr/local/mysqld\_exporter/my.cnf" &**

查看服务

netstat -antup | grep mysqld\_export

修改prometheus.yml文件，确定启动ip

- job\_name: 'mysql'

static\_configs:

- targets: ['192.168.12.129:9104']

labels:

instance: db1

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

<http://192.168.12.129:9090/targets>

grafana展示dashboard手动编辑导入

https://github.com/percona/grafana-dashboards/blob/master/dashboards/System\_Overview.json

vim System\_Overview.json

网页内容

手动导入

##### 四、Nginx-vts-exporter部署

收集nginx指标的较多采用的有nginx-vts模块，prometheus-lua两种方式进行采集，本文采用nginx-vts模块方式进行数据收集，**nginx服务本身只需要监控代理服务器，node节点nginx-vts模块status也可不部署。**

查看是否安装了nginx-vts模块

/usr/local/nginx/sbin/nginx -V

安装nginx-vts模块(如果有跳过此步骤)

cd /root/

git clone https://github.com/vozlt/nginx-module-vts #下载nginx-module-vts模块

cd /xia/software/nginx-1.19.1

./configure --prefix=/usr/local/nginx --add-module=/root/nginx-module-vts --user=nginx --group=nginx --with-stream --with-http\_ssl\_module --with-http\_stub\_status\_module

make

使用新编译的nginx执行文件替换掉旧的

mv /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.bak.20200727

cp /xia/software/nginx-1.19.1/objs/nginx /usr/local/nginx/sbin/

生成新的nginx.pid文件

kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`

退出旧版本低nginx

/usr/local/nginx/sbin/nginx.bak.20200727 -s stop

查看最新的编译参数

/usr/local/nginx/sbin/nginx -V

添加nginx配置文件,暴露监控指标(前提集群已正常工作)

vim /usr/local/nginx/conf/nginx.conf

http {

...

**vhost\_traffic\_status\_zone; #新增内容**

...

server {...

...}

**include vhost/\*.conf; #调用其他conf文件**

...

}

创建用于监控的conf文件

mkdir /usr/local/nginx/conf/vhost

vim /usr/local/nginx/conf/vhost/nginx-vts-status.conf

server {

listen 8088;

location /status {

**vhost\_traffic\_status\_display;**

**vhost\_traffic\_status\_display\_format html;**

}

}

需要重启nginx才能生效

/usr/local/nginx/sbin/nginx -s stop

/usr/local/nginx/sbin/nginx

查看网页status

http://192.168.12.129:8088/status

<http://192.168.12.129:8088/status/format/json>

http://192.168.12.129:8088/status/format/prometheus

增加status更多信息

重新配置编译

cd /xia/software/nginx-1.19.1

./configure --prefix=/usr/local/nginx --add-module=/root/nginx-module-vts --prefix=/usr/local/nginx --user=nginx --group=nginx --with-stream --with-http\_ssl\_module --with-http\_stub\_status\_module --with-http\_realip\_module --with-http\_geoip\_module

make

使用新编译的nginx执行文件替换掉旧的

mv /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.bak.20200727

cp /xia/software/nginx-1.19.1/objs/nginx /usr/local/nginx/sbin/

生成新的nginx.pid文件

kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`

退出旧版本低nginx

/usr/local/nginx/sbin/nginx.bak.20200727 -s stop

查看最新的编译参数

/usr/local/nginx/sbin/nginx -V

vim /usr/local/nginx/conf/vhost/nginx-vts-status.conf

…

listen 8088;

**vhost\_traffic\_status\_filter\_by\_set\_key $uri uri::$server\_name;** #每个uri访问量

**vhost\_traffic\_status\_filter\_by\_set\_key $geoip\_country\_code country::$server\_name;** #不同国家/区域请求量

**vhost\_traffic\_status\_filter\_by\_set\_key $status $server\_name;** #http code统计

**vhost\_traffic\_status\_filter\_by\_set\_key $upstream\_addr upstream::backend;** #后端转发统计

**vhost\_traffic\_status\_filter\_by\_set\_key $remote\_port client::ports::$server\_name;** #请求端口统计

**vhost\_traffic\_status\_filter\_by\_set\_key $remote\_addr client::addr::$server\_name;** #请求IP统计

location /status {

…

}

**location ~ ^/storage/(.+)/.\*$ {**

**set $volume $1;**

**vhost\_traffic\_status\_filter\_by\_set\_key $volume storage::$server\_name; #请求路径统计**

**}**

查看网页status

http://192.168.12.129:8088/status

<http://192.168.12.129:8088/status/format/json>

http://192.168.12.129:8088/status/format/prometheus

部署nginx-vts-exporter

tar -zxvf nginx-vts-exporter-0.10.3.linux-amd64.tar.gz

cp nginx-vts-exporter-0.10.3.linux-amd64/nginx-vts-exporter /usr/local/node\_exporter/

mv nginx-vts-exporter-0.10.3.linux-amd64/ /usr/local/src/

**/usr/local/node\_exporter/nginx-vts-exporter -nginx.scrape\_uri=http://192.168.12.129:8088/status/format/json &> /dev/null &**

查看服务

netstat -antup | grep nginx-vts-exp

修改prometheus.yml文件，确定启动ip

- job\_name: 'nginx-vts'

static\_configs:

- targets: ['192.168.12.129:9913']

labels:

instance: proxy

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

<http://192.168.12.129:9090/targets>

grafana展示dashboard手动编辑导入

wget https://grafana.com/api/dashboards/2949/revisions/1/download

手动导入

##### 五、Pushgateway部署

cd /xia/software/prometheus/

tar -zxvf pushgateway-1.2.0.linux-amd64.tar.gz -C /usr/local/

mv pushgateway-1.2.0.linux-amd64 pushgateway

起服务

**/usr/local/pushgateway/pushgateway &**

查看服务

netstat -antup | grep pushgateway

Statu-->Targets展示监控具体的监控目标

修改prometheus.yml文件，确定启动ip

- job\_name: 'push-gateway'

static\_configs:

- targets: ['192.168.12.129:9091']

labels:

instance: pushgateway

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

<http://192.168.12.129:9090/targets>

##### 六、Blackbox\_exporter部署

cd /xia/software/prometheus/

tar -zxvf blackbox\_exporter-0.17.0.linux-amd64.tar.gz -C /usr/local/

mv blackbox\_exporter-0.17.0.linux-amd64.tar.gz blackbox\_exporter

起服务

**/usr/local/blackbox\_exporter/blackbox\_exporter --config.file=/usr/local/blackbox\_exporter/blackbox.yml &**

查看服务

netstat -antup | grep blackbox\_exporter

Statu-->Targets展示监控具体的监控目标

修改prometheus.yml文件，确定启动ip

- job\_name: 'node\_status'

metrics\_path: /probe

params:

module: [icmp]

static\_configs:

- targets: ['192.168.12.129']

labels:

instance: 'master\_status'

group: 'master'

relabel\_configs:

- source\_labels: [\_\_address\_\_]

target\_label: \_\_param\_target

# - source\_labels: [\_\_param\_target]

# target\_label: instance

- target\_label: \_\_address\_\_

replacement: 192.168.12.129:9115

- job\_name: 'port\_status'

metrics\_path: /probe

params:

module: [tcp\_connect]

static\_configs:

- targets: ['192.168.12.129:8080','192.168.12.129:80','192.168.12.130:8080','192.168.12.130:80']

labels:

instance: 'port\_status'

group: 'tcp'

relabel\_configs:

- source\_labels: [\_\_address\_\_]

target\_label: \_\_param\_target

# - source\_labels: [\_\_param\_target]

# target\_label: instance

- target\_label: \_\_address\_\_

replacement: 192.168.12.129:9115

- job\_name: 'web\_status'

metrics\_path: /probe

params:

module: [http\_2xx]

static\_configs:

- targets: ['https://www.baidu.com']

labels:

instance: web\_status

group: web

relabel\_configs:

- source\_labels: [\_\_address\_\_]

target\_label: \_\_param\_target

- target\_label: \_\_address\_\_

replacement: 192.168.12.129:9115

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

<http://192.168.12.129:9090/targets>

grafana展示dashboard手动编辑导入

wget https://grafana.com/api/dashboards/9965/revisions/1/download

手动导入

#### 部署Grafana服务

cd /xia/software/

tar -xzvf grafana-7.0.6.linux-amd64.tar.gz

cd grafana-7.0.6.linux-amd64

yum -y localinstall grafana-7.0.6-1.x86\_64.rpm

配置文件

配置文件位于/etc/grafana/grafana.ini，这里暂时保持默认配置即可

设置开机启动

systemctl enable grafana-server

systemctl start grafana-server

查看服务

netstat -antup | grep :3000

添加数据

访问：http://192.168.12.129:3000，默认账号/密码：admin/admin

默认登录后，要求改密码，可以选择先跳过

添加数据源

在登陆首页，点击"Add data source"按钮，跳转到添加数据源页面，配置如下：

Name: prometheus

Type: prometheus

URL: http://192.168.12.129:9090

Access: Server(default)

取消Default的勾选，其余默认，点击"Add"，如下：

导入dashboard(只以node\_exporter为例，其他exporter类似)

从grafana官网下载相关dashboaed到本地，如：<https://grafana.com/dashboards/405>

Node\_exporter监控模板地址

<https://grafana.com/api/dashboards/8919/revisions/17/download>

Mysqld\_exporter监控模板查看

[https://github.com/percona/grafana-dashboards/blob/master/dashboards/System\_Overview.json](https://github.com/percona/grafana-dashboards/blob/master/dashboards/System_Overview.json）)/

Nginx\_exporter监控模板地址

https://grafana.com/api/dashboards/2949/revisions/2/download

Grafana首页-->左上角图标-->Dashboard-->import

加入node-exporter-0-16-0-17-for-prometheus\_rev7.json这个插件

如果出现grafana-piechart-panel这个报错

1. 安装pie插件

官网：https://grafana.net/plugins/grafana-piechart-panel

grafana-cli plugins install grafana-piechart-panel # pmm-singlestat-panel grafana-piechart-panel #可能因为网络原因需多次执行

2）重启grafana-server #如果源码安装: killall grafana-server ./grafana-server restart &

systemctl restart grafana-server.service

3）点击导入的dashboard查看效果

4) 增删改查增加某个监控服务的监控项

#### 部署Alertmanager服务

以下规则定义当任务node某一主机node\_exporter服务挂掉，即产生一个告警

vim /usr/local/prometheus/rule.yml

groups:

- name: alert-rules #告警的分组，后续告警优化时，可做优化配置

rules:

- alert: InstanceStatus #告警规则名称

expr: up{job="export\_test2"} == 0 #1是服务正常 ，0服务挂了

for: 10s #评估等待10s，等待期间报警状态为pending

labels: #此标签可用于match之后的通知操作

severity: 'critical'

annotations: #描述告警信息

description: 服务器 已宕机

summary: 服务器 运行状态

在prometheus的配置文件中添加该规则

vim /usr/local/prometheus/prometheus.yml

rule\_files:

- "/usr/local/prometheus/rule.yml"

下载插件

wget https://github.com/prometheus/alertmanager/releases/download/v0.15.2/alertmanager-0.15.2.linux-amd64.tar.gz

安装

tar -xvf alertmanager-0.15.2.linux-amd64.tar.gz -C /usr/local

mv /usr/local/alertmanager-0.15.2.linux-amd64 /usr/local/alertmanager

起服务

cd /usr/local/alertmanager-0.15.2

./alertmanager &

在prometheus的配置文件中配置alertmanager地址，让其知晓alertmanager的地址，以传送告警信息

查看服务

netstat -antup | grep alertmanager

修改prometheus.yml文件，确定启动ip

vim /usr/local/prometheus/prometheus.yml

alerting:

alertmanagers:

- static\_configs:

- targets: ['192.168.12.129:9093']

修改完后重启prometheus

killall -9 prometheus

/usr/local/prometheus/prometheus --config.file="/usr/local/prometheus/prometheus.yml" &

网页查看

http://192.168.12.129:9090/rules

http://192.168.12.129:9090/alerts

测试：  
此时关闭192.168.20.137的node\_exporter服务

killall node\_exporter

查看prometheus的web界面，已经产生告警：

http://192.168.12.129:9090/alerts

再查看alertmanager已经接收到来自prometheus的告警：

http://192.168.12.129:9093/#/alerts

配置告警处理策略

新建alertmanager.yml,由于要进行发邮件测试，将等待时间和间隔时间设置的很短

vim /usr/local/alertmanager/alertmanager.yml

global:

resolve\_timeout: 5m

smtp\_smarthost: 'smtp.qq.com:25'

smtp\_from: '1191246623@qq.com'

smtp\_auth\_username: '1191246623@qq.com'

smtp\_auth\_password: 'gotbxagaxschheff' #这是授权码，不是密码，

smtp\_require\_tls: false

route:

group\_by: ['alertname']

group\_wait: 2s

group\_interval: 8s

repeat\_interval: 20h

receiver: 'default-receiver'

routes:

- receiver: 'default-receiver'

match: #匹配告警规则的key:value

severity: 'critical'

receivers:

- name: 'default-receiver'

email\_configs:

- to: 'xia\_xingming@163.com'

send\_resolved: true #告警解除发送恢复通知

增删改查增加某个监控服务的rule和altermanager文件

#### 批量部署

1. 批量示范(以ansible部署prometheus为例)



1. 服务配置相关操作
2. 配置文件修改(Hostname根据agent实际情况更改)

vim /usr/local/zabbix/etc/zabbix\_agentd.conf

Server=127.0.0.1

ServerActive=192.168.12.129

Hostname=Agent1

StartAgents=0 #禁止使用被动模式

RefreshActiveChecks=120

1. 服务启动

cd /etc/ansbile

ansible all -a 'systemctl start mysqld;/usr/local/nginx/sbin/nginx;/usr/local/zabbix/sbin/zabbix\_agentd;systemctl start php-fpm;systemctl start httpd'

1. 服务状态查看

netstat -antup | grep nginx

netstat -antup | grep zabbix\_agent

1. 监控脚本及配置文件验证

如果脚本和配置文件无误，可以得到访问结果

/usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k nginx.status[active]

/usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k nginx.status[reading]

1. 网页配置

参考zabbix\_server网页配置

1. 图像查看

查看zabbix\_agent监控数据