链滴

Prometheus+Grafana+Zabbix+alertmanager 部署

作者：GeekBoyDqz

原文链接：https://ld246.com/article/1571056043506

来源网站：链滴

<h2 id="一-准备环境">一、准备环境</h2>
<table>
<thead>
<tr>
<th>程序</th>
<th>OS</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>prometheus& grafana server</td>
<td>centos 7.6</td>
<td>192.168.66.52</td>
</tr>
<tr>
<td>prometheus node</td>
<td>centos 7.6</td>
<td>192.168.66.53</td>
</tr>
</tbody>
</table>
<h2 id="二-部署prometheus">二、部署 prometheus</h2>
在 prometheus& grafana server 节点部署 prometheus 服务。
<h3 id="1--下载-部署">1. 下载&部署</h3>

下载，官网：<a href="https://ld246.com/forward?goto=https%3A%2F%2Fprometheus.io
2F" target="_blank" rel="nofollow ugc">https://prometheus.io/

<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# mkdir /soft
[root@prometheu
~]# cd /soft/

[root@prometheu
soft]# wget https://github.com/prometheus/prometheus/releases/download/v2.12.0/promet
eus-2.12.0.linux-amd64.tar.gz

#部署到/usr/local
目录
#promethus不用
译安装，解压目录中有配置文件与启动文件
[root@prometheu
soft]# tar -zxvf prometheus-2.12.0.linux-amd64.tar.gz -C /usr/local/

[root@prometheu
soft]# cd /usr/local/
[root@prometheu
local]# mv prometheus-2.12.0.linux-amd64 prometheus

#验证
[root@prometheu
local]# cd prometheus/
[root@prometheu

prometheus]# ./prometheus --version
</code></pre>
<blockquote>
<p><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2019/10/image-cce81448.png?imageView2/2/interlace/1/format/jpg"></>
</blockquote>
<h3 id="2--设置用户">2. 设置用户</h3>
<pre><code class="highlight-chroma"># 添加用户，后期用此账号启动服务
[root@prometheu prometheus]# groupadd prometheus
[root@prometheu prometheus]# useradd -g prometheus -s /sbin/nologin prometheus

赋权
[root@prometheu prometheus]# cd ~
[root@prometheu ~]# chown -R prometheus:prometheus /usr/local/prometheus/

创建prometheu运行数据目录
[root@prometheu ~]# mkdir -p /var/lib/prometheus
[root@prometheu ~]# chown -R prometheus:prometheus /var/lib/prometheus/
</code></pre>
<h3 id="3--设置开机启动">3. 设置开机启动</h3>
<pre><code class="highlight-chroma">[root@prometheus ~]# vim /etc/systemd/system/prometheus.service

[Unit]
Description=Prometheus
Documentation=https://prometheus.io/
After=network.target

[Service]
# Type设置为notif时，服务会不断重启
Type=simple
User=prometheus
# --storage.tsdb.pth是可选项，默认数据目录在运行目录的./dada目录中
ExecStart=/usr/local/prometheus/prometheus --config.file=/usr/local/prometheus/prometheus.yml --storage.tsb.path=/var/lib/prometheus
Restart=on-failure

[Install]
WantedBy=multi-

ser.target
</code></pre>

设置开机启动

<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# systemctl daemon-reload
[root@prometheu
~]# systemctl enable Prometheus
[root@prometheu
~]# systemctl start prometheus
</code></pre>
<h3 id="4--设置iptables或Firewalld">4. 设置 iptables 或 Firewalld</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# vim /etc/sysconfig/iptables
-A INPUT -p tcp
m state --state NEW -m tcp --dport 9090 -j ACCEPT

[root@prometheu
~]# service iptables restart

#centos7默认开启
irewalld
查看目前开放的端

[root@prometheu
~]# firewall-cmd --zone=public --list-ports

开放9090端口
[root@prometheu
~]# firewall-cmd --permanent --zone=public --add-port=9090/tcp #永久添加

使其规则生效
[root@prometheu
~]# firewall-cmd --reload
</code></pre>
<h3 id="5--启动并验证">5. 启动并验证</h3>
<h4 id="1-查看服务状态">1）查看服务状态</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# systemctl status prometheus
[root@prometheu
~]# ss -tunlp | grep 9090
</code></pre>
<h4 id="2-web-ui">2）web ui</h4>
<blockquote>
Prometheus 自带有简单的 UI，<a href="https://ld246.com/forward?goto=http%3A%2F%
F192.168.66.52%3A9090" target="_blank" rel="nofollow ugc">http://192.168.66.52
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-29d09f7a.png?imageView2/2/interlace/1/format/jpg">
</blockquote>
<blockquote>
在 Status 菜单下，Configuration，Rule，Targets 等，
Statu-->Configuration 展示 prometheus.yml 的配置，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3

ogfile.com/file/2019/10/image-691dc50e.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-838a45e8.png?imageView2/2/interlace/1/format/jpg">
</blockquote>
<blockquote>
Statu-->Targets 展示监控具体的监控目标，这里监控目标"linux"暂未设置 node_exporter
未 scrape 数据，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-288445fd.png?imageView2/2/interlace/1/format/jpg">
</blockquote>
<h3 id="7--绘图">7. 绘图</h3>

访问：<a href="https://ld246.com/forward?goto=http%3A%2F%2F192.168.66.52%3A909
" target="_blank" rel="nofollow ugc">http://192.168.66.52/metrics，查看从 exporter 具
能抓到的数据，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-77d1309c.png?imageView2/2/interlace/1/format/jpg">

访问：<a href="https://ld246.com/forward?goto=http%3A%2F%2F192.168.66.52%3A909
" target="_blank" rel="nofollow ugc">http://192.168.66.52，在输入框中任意输入 1 个 exp
rter 能抓取得值，点击"Execute"与"Execute"按钮，即可见相应抓取数据的图形，同时可对时间与 uni
做调整，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-d7937b9b.png?imageView2/2/interlace/1/format/jpg">

<h2 id="三-部署node-exporter">三、部署 node_exporter</h2>

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

这里在 prometheus node 节点部署相关服务。

<h3 id="1--下载-部署-">1. 下载&部署</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl"># 下载
[root@node1 ~]#
d /soft
[root@node1 soft
wget https://github.com/prometheus/node_exporter/releases/download/v0.15.1/node_exp
rter-0.18.1.linux-amd64.tar.gz

#部署
[root@node1 soft
tar -zxvf node_exporter-0.18.1.linux-amd64.tar.gz -C /usr/local/
[root@node1 soft
cd /usr/local
[root@node1 local
mv node_exporter-0.18.1.linux-amd64 node_exporter
</code></pre>
<h3 id="2--设置用户-">2. 设置用户</h3>
<pre><code class="highlight-chroma"><span class="highlight

cl">[root@node1 ~]# groupadd prometheus
[root@node1 ~]#
useradd -g prometheus -s /sbin/nologin prometheus
[root@node1 ~]#
hown -R prometheus:prometheus /usr/local/node_exporter/
</code></pre>
<h3 id="3--设置开机启动-">3. 设置开机启动</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@node1 ~]# vim /etc/systemd/system/node_exporter.service
[Unit]
Description=node
exporter
Documentation=h
tps://prometheus.io/
After=network.tar
et

[Service]
Type=simple
User=prometheus
ExecStart=/usr/lo
al/node_exporter/node_exporter
Restart=on-failure

[Install]
WantedBy=multi-
ser.target

[root@node1 ~]#
ystemctl daemon-reload
[root@node1 ~]#
ystemctl enable node_exporter
[root@node1 ~]#
ystemctl start node_exporter
</code></pre>
<h3 id="4--设置iptables或Firewalld-">4. 设置 iptables 或 Firewalld</h3>
<pre><code class="highlight-chroma">>><span class="highlight
cl">># 官方node_exporter默认使用9100端口
>[root@node1 ~]#
im /etc/sysconfig/iptables
>-A INPUT -p tcp
m state --state NEW -m tcp --dport 9100 -j ACCEPT
>
>[root@node1 ~]#
ervice iptables restart
>
>># centos7默认开启
irewalld
>查看目前开放的端

>[root@node1 ~]#
irewall-cmd --zone=public --list-ports
>

开放9100端口
[root@node1 ~]# irewall-cmd --permanent --zone=public --add-port=9100/tcp #永久添加

使其规则生效
[root@node1 ~]# irewall-cmd --reload

再次查看端口
[root@node1 ~]# irewall-cmd --zone=public --list-ports

#检查服务状态
[root@node1 ~]# ystemctl status node_exporter
[root@node1 ~]# ss -tnlp | grep 9100
</code></pre>
<h3 id="5--Prometheus添加该主机">5. Prometheus 添加该主机</h3>
<pre><code class="highlight-chroma">[root@prometheus ~]# vim /usr/local/prometheus/prometheus.yml
#简单验证，主要置采用默认文件配置，有修改/新增处用红色标示
[root@prometheu prometheus]# vim prometheus.yml
#全局配置
global:
  scrape_interval: 15s # 设置抓取(pull)时间间隔，默认是1m
  evaluation_interal: 15s # 设置rules评估时间间隔，默认是1m
  # scrape_timeout is set to the global default (10s).

#告警管理配置，未使用，默认配置
alerting:
  alertmanagers:
  - static_configs:
    - targets:
      # - alertmaner:9093

#加载rules，并根设置的时间间隔定期评估，暂未使用，默认配置
rule_files:
  # - "first_rules.ym"
  # - "second_rules yml"

#抓取(pull)，即控目标配置
#默认只有主机本

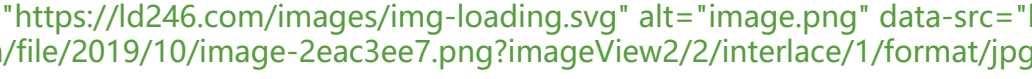
的监控配置
scrape_configs:

  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ['localhost:9090']

- job_name: 'linux'
    static_configs:
    - targets: ['192.168.66.53:9100'] #被监控端服务器ip
      labels:
        instance: node1
#监控目标的label这里的监控目标只是一个metric，而不是指某特定主机，可以在特定主机取多个监控目标），在抓取每条时间序列表中都会添加此label
</code></pre>

注意：job_name 定义的组名必须不能相同


<pre><code class="highlight-chroma">#重启prometheus服务
[root@prometheus ~]# systemctl restart prometheus
</code></pre>
<h3 id="6-验证">6、验证</h3>

访问：http://192.168.66.52，可见 node1 主机已经可被监，如下：


<h2 id="四-部署grafana">四、部署 grafana</h2>

在 prometheus& grafana server 节点部署 grafana 服务。

<h3 id="1--下载-安装">1. 下载&安装</h3>
<pre><code class="highlight-chroma">https://grafana.com/grafana/download
#下载，可以下载之后上传至服务器
[root@prometheus ~]# cd /usr/local/src/
[root@prometheu

soft]# wget https://dl.grafana.com/oss/release/grafana-6.3.3-1.x86_64.rpm

安装本地rpm包 l
calinstall
[root@prometheu
soft]# yum localinstall grafana-6.3.3-1.x86_64.rpm
</code></pre>

<h3 id="2--配置文件">2. 配置文件</h3>

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

<h3 id="3--设置开机启动--">3. 设置开机启动</h3>

<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus soft]# systemctl enable grafana-server
[root@prometheu
soft]# systemctl start grafana-server

#查看服务状态
[root@prometheu
soft]# systemctl status grafana-server
[root@prometheu
soft]# ss -tnlp | grep 3000
</code></pre>

<h3 id="4--设置iptables">4. 设置 iptables</h3>

<pre><code class="highlight-chroma"><span class="highlight
cl"># grafana-server默认使用3000端口
[root@prometheu
soft]# vim /etc/sysconfig/iptables
-A INPUT -p tcp
m state --state NEW -m tcp --dport 3000 -j ACCEPT

[root@prometheu
soft]# service iptables restart

centos7默认开启
irewalld
查看目前开放的端

[root@prometheu
soft]# firewall-cmd --zone=public --list-ports

开放3000端口
[root@prometheu
soft]# firewall-cmd --permanent --zone=public --add-port=3000/tcp   #永久添加

使其规则生效
[root@prometheu
soft]# firewall-cmd --reload

再次查看端口
[root@prometheu
soft]# firewall-cmd --zone=public --list-ports
</code></pre>

<h3 id="5--添加数据源">5. 添加数据源</h3>

<h4 id="1-登陆">1）登陆</h4>

访问：http://192.168.66.52，默认账号/密码：admin/admin

<h4 id="2-添加数据源">2）添加数据源</h4>

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：

<blockquote>
Name: prometheus
Type: prometheus
URL: http://localhost:9090/　　#prometheus 服务器所在地址，如果就在本机填写 localhost

Access: Server
</blockquote>

其余默认，点击"Add"，如下：

在"Dashboards"页签下"import"自带的模版，如下：

<h3 id="6--导入dashboard">6. 导入 dashboard</h3>

从 grafana 官网下载相关 dashboaed 到本地，如：https://grafana.co/

Grafana 首页--> 页面顶部-->Dashboards--> 页面左边 Filter by-选择仪表盘 prometheus

ogfile.com/file/2019/10/image-a121ccc5.png?imageView2/2/interlace/1/format/jpg">

上传已下载至本地的 json 文件到 Grafana 服务器[192.168.66.52:3000]（或者使用 dashboard），如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-c56bc036.png?imageView2/2/interlace/1/format/jpg">

数据源选择"prometheus"，即添加的数据源 name，点击"Import"按钮，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-1ad700d5.png?imageView2/2/interlace/1/format/jpg">

<h3 id="7--安装饼图的插件">7. 安装饼图的插件</h3>
<pre><code class="highlight-chroma">#使用新的grafana-cli工具从命令行安装piechart-panel：
[root@prometheu ~]# grafana-cli plugins install grafana-piechart-panel

该插件将安装到您grafana插件目录中; 如果安装了grafana软件包，则默认在/var/lib/grafana/plugins
[root@prometheu ~]# cd /var/lib/grafana/plugins/

#重启grafana
[root@prometheu ~]# systemctl restart grafana-server

这样dashboard中饼图就可以正常展示出来了
</code></pre>
<h3 id="8--查看dashboard">8. 查看 dashboard</h3>

Grafana 首页--> 左上角图标--> Home，Home 列表中可见有已添加的两个 dashboard " Node Exporter 0.16 0.17 for Prometheus 监控展示看板"与"Node dashboard Copy"，选择 1 即可，如下：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-5e987f52.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-fb197bce.png?imageView2/2/interlace/1/format/jpg">

<h3 id="9--更新dashboard名称">9.　更新 dashboard 名称</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2019/10/image-9feac7ac.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-9318c6f8.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-7a6689e7.png?imageView2/2/interlace/1/format/jpg">
<h3 id="10--删除dashboard操作">10.　删除 dashboard 操作</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:

/b3logfile.com/file/2019/10/image-f601f439.png?imageView2/2/interlace/1/format/jpg"><b
>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-9065cd11.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-9a80515a.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3
ogfile.com/file/2019/10/image-a7c6e413.png?imageView2/2/interlace/1/format/jpg">
<h2 id="五-部署mysql-exporter">五、部署 mysql_exporter</h2>
<h3 id="1-部署的架构图">1、部署的架构图</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2019/10/image-267c448e.png?imageView2/2/interlace/1/format/jpg"></
>
<h3 id="2-安装mysqld-exporter">2、安装 mysqld_exporter</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">在node节点安装
[root@node2 ~]#
wget https://github.com/prometheus/mysqld_exporter/releases/download/v0.11.0/mysqld_e
porter-0.11.0.linux-amd64.tar.gz

[root@node2 ~]#
ar xzvf mysqld_exporter-0.11.0.linux-amd64.tar.gz -C /usr/local/

[root@node2 ~]#
d /usr/local/

[root@node2 local
mv mysqld_exporter-0.11.0.linux-amd64/ mysqld_exporter
</code></pre>
<h3 id="3-设置账户">3、设置账户</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@node2 ~]# groupadd prometheus

[root@node2 ~]#
useradd -g prometheus -s /sbin/nologin prometheus

#更改目录权限
[root@node2 ~]#
hown -R prometheus:prometheus /usr/local/mysqld_exporter/
</code></pre>
<h3 id="4-添加MySQL远程登入用户">4、添加 MySQL 远程登入用户</h3>

注意：MySQL5.7 设置密码时必须符合长度，与 validate_password_policy 的值有关。如果要
改简单点密码可以设置以下参数值

<pre><code class="highlight-chroma"><span class="highlight
cl">mysql> set global validate_password_policy=0;

mysql> GRANT
REPLICATION CLIENT, PROCESS ON *.* TO 'mysqld_exporter'@'%' identified by '123456';

mysql> GRANT
SELECT ON performance_schema.* TO 'mysqld_exporter'@'%';

mysql> flush pr
vileges;
</code></pre>
<h3 id="5-创建一个用于连接MySQL的配置文件">5、创建一个用于连接 MySQL 的配置文件</h3>

mysqld_exporter 默认会读取~/.my.cnf 文件。这里是创建在 mysqld_exporter 的安装目录下

<pre><code class="highlight-chroma"><span class="highlight
cl">[root@node2 ~]# vim /usr/local/mysqld_exporter/.my.cnf
[client]
user=mysqld_exp
rter //刚才创建的用户
password=123456
  //登入密码
</code></pre>
<h3 id="6-创建systemd服务">6、创建 systemd 服务</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@exporter ~]# vim /etc/systemd/system/mysqld_exporter.service

[Unit]
Description=mysq
_exporter
After=network.tar
et
[Service]
Type=simple
User=prometheus
ExecStart=/usr/lo
al/mysqld_exporter/mysqld_exporter --config.my-cnf=/usr/local/mysqld_exporter/.my.cnf
Restart=on-failure
[Install]
WantedBy=multi-
ser.target
</code></pre>
<h3 id="7-启动mysqld-exporter">7、启动 mysqld_exporter</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@node2 ~]# systemctl enable mysqld_exporter

[root@node2 ~]#
ystemctl start mysqld_exporter

检查端口9104
[root@node2 ~]#
sof -i:9104
COMMAND PID
  USER FD TYPE DEVICE SIZE/OFF NODE NAME
mysqld_ex 2573 p
ometheus 3u IPv6 23391639 0t0 TCP *:peerwire (LISTEN)
</code></pre>
<h3 id="8-设置iptables或者Firewalld">8、设置 iptables 或者 Firewalld</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">#官方mysqld_exporter默认使用9104端口
[root@node1 ~]#

im /etc/sysconfig/iptables
-A INPUT -p tcp
m state --state NEW -m tcp --dport 9104 -j ACCEPT

[root@node2 ~]#
ervice iptables restart

centos7默认开启
irewalld
查看目前开放的端

[root@node2 ~]#
irewall-cmd --zone=public --list-ports

开放9104端口
[root@node2 ~]#
irewall-cmd --permanent --zone=public --add-port=9104/tcp   #永久添加

使其规则生效
[root@node2 ~]#
irewall-cmd --reload

再次查看端口
[root@node2 ~]#f
rewall-cmd --zone=public --list-ports
</code></pre>
<h3 id="9-修改prometheus-yml-加入下面的监控目标">9、修改 prometheus.yml，加入下面的
控目标</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# vim /usr/local/prometheus/prometheus.yml
  - job_name: mys
l
    static_configs:
      - targets: ['192
168.66.11:9104']
        labels:
          instance: d
1
</code></pre>
<h3 id="8-重启Prometheus">8、重启 Prometheus</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@Prometheus ~]# systemctl restart prometheus
</code></pre>
<h3 id="9-在Grafana中导入模板">9、在 Grafana 中导入模板</h3>
<h4 id="9-1-上传文件">9.1:上传文件</h4>

点击页面左侧---Dashboards---Home---即可看见刚才创建的仪表盘

<h2 id="六-Grafana配合zabbix展示">六、Grafana 配合 zabbix 展示</h2>
<h3 id="1-安装zabbix插件-在Grafana服务器">1、安装 zabbix 插件，在 Grafana 服务器</h3>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">[root@prometheus ~]# grafana-cli plugins install alexanderzobnin-zabbix-app
重启Grafan
[root@prometheu

~]# systemctl restart grafana-server
</code></pre>
<h3 id="2-添加数据源-">2、添加数据源</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2019/10/image-8a3e219f.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-462be6ae.png?imageView2/2/interlace/1/format/jpg">
<blockquote>
URL http://zabbix 服务 IP/zabbix/api_jsonrpc.php
Zabbix API details
Username：zabbix 登入用户
Password：zabbix 用户密码
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-97690712.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-de69945f.png?imageView2/2/interlace/1/format/jpg">
</blockquote>

根据安装的 zabbix 选择对于的版本
点击 Save & Test 保存
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-ace2b75d.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-976d7fe8.png?imageView2/2/interlace/1/format/jpg">

<h3 id="3-创建数据源">3、创建数据源</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:/b3logfile.com/file/2019/10/image-b97d85c9.png?imageView2/2/interlace/1/format/jpg"></>
<blockquote>
1、首先选择数据源：zabbix
2、选择 zabbix 内的主机组
3、选择主机
4、选择应用集
5、选择监控项
6、给这一条数据添加一个名称（将显示在图表下方，方便观看者辨识每条曲线代表什么意思）
7、如果需要添加多条曲线点击 ADD Query，然后循环 2-6 即可
8、完成后点击右边的白色叉叉即可展现出图表，如下图：
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-07f68d5e.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-c5f03579.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-79341589.png?imageView2/2/interlace/1/format/jpg">
</blockquote>

右上角点击保存图标进行保存
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-617b2dbd.png?imageView2/2/interlace/1/format/jpg">

定义 dashboard 名称

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-a795a9f2.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-333594c6.png?imageView2/2/interlace/1/format/jpg">

<h2 id="七-Alertmanager告警">七、Alertmanager 告警</h2>

首先需要在企业微信中添加应用小程序
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-38e585d8.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-17cc3dce.png?imageView2/2/interlace/1/format/jpg">
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-da781439.png?imageView2/2/interlace/1/format/jpg">

应用的 ID 需要记住，后面使用
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-d44195c0.png?imageView2/2/interlace/1/format/jpg">

企业的唯一 ID 需要记住，后面使用
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-50824b4d.png?imageView2/2/interlace/1/format/jpg">

<h3 id="1-安装Alertmanager">1、安装 Alertmanager</h3>

https://prometheus.io/download/
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-54362414.png?imageView2/2/interlace/1/format/jpg">

<pre><code class="highlight-chroma">[root@prometheus ~]# tar -xzvf alertmanager-0.18.1.linux-amd64.tar.gz –C /usr/local/
[root@prometheu local]# cd /usr/local/

[root@prometheu local]# mv alertmanager-0.18.0.linux-amd64/ alertmanager
</code></pre>
<h3 id="2-配置prometheus-yml">2、配置 prometheus.yml</h3>
<pre><code class="highlight-chroma">

[root@prometheus ~]# vim /usr/local/prometheus/prometheus.yml

![image.png](https//b3logfile.com/file/2019/10/image-abf214bb.png)
</code></pre>
<h3 id="3-创建告警规则文件">3、创建告警规则文件</h3>
<pre><code class="highlight-chroma">

[root@prometheus ~]# cd /usr/local/prometheus/

[root@prometheu prometheus]# vim rules.yml

groups:
- name: node
  rules:
  - alert: server_status # alert 名字
    expr: up{job="prometheus"} == 0   #job是prometheus.yml中的jobname
    for: 15s   # 条保持 15s 才会发出 alert
    labels: #自定标签
      severity: page
annotations: # alert 的其他标签，但不用于标识 alert
      ummary: "机器 {{ $labels.instance }} 挂了"   #告警标题
      description: "立即查看"   #告警详细内容
</code></pre>
<blockquote>
alert：告警规则的名称。
expr：基于 PromQL 表达式告警触发条件，用于计算是否有时间序列满足该条件。
for：评估等待时间，可选参数。用于表示只有当触发条件持续一段时间后才发送告警。在等待期间新生告警的状态为 pending。
labels：自定义标签，允许用户指定要附加到告警上的一组附加标签。
annotations：用于指定一组附加信息，比如用于描述告警详细信息的文字等，annotations 的内容告警产生时会一同作为参数发送到 Alertmanager。
</blockquote>
<h3 id="4-配置alertmanager-yml">4、配置 alertmanager.yml</h3>
<pre><code class="highlight-chroma">[root@prometheus prometheus]# cd /usr/local/alertmanager/
[root@prometheus alertmanager]# vim alertmanager.yml
global:
  resolve_timeout: 5m #该参数定义了当Alertmanager持续多长时间未接收到告警后标记告警状态为resolved（已解）。该参数的定义可能会影响到告警恢复通知的接收时间，读者可根据自己的实际场景进行定义，其认值为5分钟

templates: #定义告警模板文件
- '/usr/local/alertmanager/wechat.tmpl'

route:
  group_by: ['alertname'] #将传入的报警中有这些标签的分为一个组.
  group_wait: 10s   #指分组创建多久后才可以发送压缩的警报，也就是初次发警报的延时.这样会确保第一次通知的时候 有更多的报警被压缩在一起.
  group_interval: 0s #当第一个通知发送，等待多久发送压缩的警报
  repeat_interval: 1h #如果报警发送成功, 等待多久重新发送一次

    receiver: 'wecha
'   #默认的接收器
receivers:
- name: 'wechat'
  wechat_configs:
  - corp_id: 'xxx'
    to_party: '1'
    agent_id: '1000
02'
    api_secret: 'xxx'
    send_resolved: t
ue
</code></pre>
<blockquote>
corp_id: 企业微信账号唯一 ID， 可以在我的企业中查看。
to_party: 需要发送的组。
agent_id: 第三方企业应用的 ID，可以在自己创建的第三方企业应用详情页面查看。
api_secret: 第三方企业应用的密钥，可以在自己创建的第三方企业应用详情页面查看。
</blockquote>
<h3 id="5-配置wechat告警模板">5、配置 wechat 告警模板</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@prometheus ~]# vim /usr/local/alertmanager/wechat.tmpl

{{ define "wechat.
efault.message" }}
{{ range .Alerts }}
========start
=========
告警程序：prome
eus_alert
告警级别：{{ .Label
.severity }}
告警类型：{{ .Label
.alertname }}
故障主机: {{ .Labels.
nstance }}
告警主题: {{ .Annot
tions.summary }}
告警详情: {{ .Annot
tions.description }}
触发时间: {{ .Starts
t.Format "2006-01-02 15:04:05" }}
========end=
========
{{ end }}
{{ end }}
</code></pre>
<h3 id="6-配置systemd启动">6、配置 systemd 启动</h3>
<pre><code class="highlight-chroma">

<span class="highlight
cl">[root@prometheus ~]# vim /etc/systemd/system/alertmanager.service

[Unit]

Description=Pro
etheus: the alerting system

Documentation=h

tp://prometheus.io/docs/
After=prometheus service

[Service]
ExecStart=/usr/local/alertmanager/alertmanager --config.file=/usr/local/alertmanager/alertmanager.yml
Restart=always
StartLimitInterval 0
RestartSec=10

[Install]
WantedBy=multi-ser.target
</code></pre>
<h3 id="7-启动Alertmanager">7、启动 Alertmanager</h3>
<pre><code class="highlight-chroma">[root@prometheus ~]# systemctl start alertmanager
[root@prometheu ~]# systemctl enable alertmanager
</code></pre>
###8、检查启动状态
<pre><code class="highlight-chroma">[root@prometheus ~]# ss –tnlp | grep 9093
</code></pre>

注意：如果启动检查没有 9093 端口，那么需要查看系统日志
如果报错为：did not find expected key
原因：alertmanager.yml 的配置文件有问题，出现了空格或者格式不对导致
解决方案：检查配置文件进行修改

<h3 id="9-浏览器访问查看">9、浏览器访问查看</h3>

查看 prometheus 页面

查看 Alertmanager 页面
http://ip:9093

<h3 id="10-手动停止node-exporter测试">10、手动停止 node_exporter 测试</h3>
<pre><code class="highlight-chroma">[root@prometheus ~]# systemctl stop node_exporte
</code></pre>
<h3 id="11-prometheus监控页面查看Alert">11、prometheus 监控页面查看 Alert</h3>

<b>

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3ogfile.com/file/2019/10/image-ec3af4cc.png?imageView2/2/interlace/1/format/jpg">
<h3 id="12-查看AlertManager告警页面">12、查看 AlertManager 告警页面</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2019/10/image-648c446e.png?imageView2/2/interlace/1/format/jpg"></>
<h3 id="13-查看微信消息">13、查看微信消息</h3>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2019/10/image-09b6c27c.png?imageView2/2/interlace/1/format/jpg"></>
<h3 id="14-常用告警规则总结">14、常用告警规则总结</h3>
<blockquote>
告警规则不会定义，也可以在 Grafan 中直接复制即可
可以在一个告警规则文件中定义多个告警项
</blockquote>
<h3 id="1-修改prometheus配置文件">1、修改 prometheus 配置文件</h3>
<pre><code class="highlight-chroma">[root@prometheus ~]# cd /usr/local/prometheus/
[root@prometheu prometheus]# vim prometheus.yml
</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2019/10/image-5f5335ed.png?imageView2/2/interlace/1/format/jpg"></>
<h3 id="2-创建目录和告警规则文件">2、创建目录和告警规则文件</h3>
<pre><code class="highlight-chroma">[root@prometheus prometheus]# mkdir rules
[root@prometheu prometheus]# chown -R prometheus. rules

[root@prometheu prometheus]# cd rules/
[root@prometheu rules]# vim cpu_rules.yml
groups:
- name: hostStatslert
  rules:
  - alert: hostCpuUageAlert
    expr: sum(avg wthout (cpu)(irate(node_cpu{mode!='idle'}[5m]))) by (instance) > 0.85
    for: 1m
    labels:
      severity: page
    annotations:
      summary: "Instance {{ $labels.instance }} CPU usgae high"
      description: "{{ $labels.instance }} CPU usage above 85% (current value: {{ $value }})"
  - alert: hostMeUsageAlert
    expr: (node_memory_MemTotal - node_memory_MemAvailable)/node_memory_MemTotal > 0.85

    for: 1m
    labels:
      severity: page
    annotations:
      summary: "Instance {{ $labels.instance }} MEM usgae high"
      description: "{{ $labels.instance }} MEM usage above 85% (current value: {{ $value }})"
</code></pre>
<h3 id="3-重启prometheus">3、重启 prometheus</h3>
<pre><code class="highlight-chroma">[root@prometheus rules]# systemctl restart prometheus
</code></pre>
<h3 id="4-查看页面">4、查看页面</h3>
<p><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2019/10/image-72174a8f.png?imageView2/2/interlace/1/format/jpg"></>
<h3 id="5-测试告警">5、测试告警</h3>

修改告警规则文件触发告警
