

用 Prometheus 细化 Nginx 监控

国内用 Nginx 的比较多，Nginx 的监控比较老的方案可能是通过跑脚本定期收集 nginx 的 status 模块的数据，或者监控 nginx 的日志；后来阿里的 **tengine** 在国内开始流行，于是诞生了很多不错的 lua 模块；但是这些监控方案在有新的监控需求的时候，可能就需要再修改脚本或者更改 nginx conf 配置，有时候不是特别的方便。用 Prometheus 进行 nginx 的监控可以自动的对相关 **server_name** 和 **upstream** 进行监控，你也可以自定义 Prometheus 的数据标签，实现对不同机房和不同项目的 nginx 进行监控。

监控 Nginx 主要用到以下三个模块：

nginx-module-vts: Nginx virtual host traffic status module，Nginx 的监控模块，能够提供 JSON 格式的数据产出。

nginx-vts-exporter: Simple server that scrapes Nginx vts stats and exports them via HTTP for Prometheus consumption。主要用于收集 Nginx 的监控数据，并给 Prometheus 提供监控接口，默认端口号 9913。

Prometheus: 监控 Nginx-vts-exporter 提供的 Nginx 数据，并存储在时序数据库中，可以使用 PromQL 对时序数据进行查询和聚合。

一、nginx-module-vts 模块的编译

nginx_vts_exporter 依赖 nginx-module-vts 模块，安装此模块无需任何其他依赖。模块与 Nginx 的版本兼容性如下：

1.11.x (last tested: 1.11.10)

1.10.x (last tested: 1.10.3)

1.8.x (last tested: 1.8.0)

1.6.x (last tested: 1.6.3)

1.4.x (last tested: 1.4.7)

同时适用于 tengine，其他 nginx 早期版本未做验证。

安装步骤：

1. 下载模块

```
shell> git clone  
git://github.com/vozlt/nginx-module-vts.git
```

2.

编译配置

在 nginx 编译时添加 vts 模块

```
--add-module=/path/to/nginx-module-vts
```

下载官方的软件包并编译进 vts 模块，例如：

3.

```
./configure --user=www --group=www --prefix=/usr/local/nginx  
--with-http_sysguard_module --add-module=nginx-module-vts
```

- 4.
- 5.

安装:

```
make && make install
```

二、Nginx Conf 配置

更改 Nginx Conf 的配置，添加监控接口/status/:

- 6.

```
http {  
    vhost_traffic_status_zone;  
  
    vhost_traffic_status_filter_by_host on;  
  
    ...  
  
    server {  
  
        ...  
  
        location /status {  
            vhost_traffic_status_display;  
            vhost_traffic_status_display_format html;  
        }  
    }  
}
```

- 7.

配置建议:

1. 打开 vhost 过滤:

```
vhost_traffic_status_filter_by_host on;
```

开启此功能，在 Nginx 配置有多个 `server_name` 的情况下，会根据不同的 `server_name` 进行流量的统计，否则默认会把流量全部计算到第一个 `server_name` 上。

2. 在不想统计流量的 `server` 区域禁用 `vhost_traffic_status`，配置示例:

```
server {  
  
    ...  

```

```
vhost_traffic_status off;  
...  
}
```

3.

假如 nginx 没有规范配置 `server_name` 或者无需进行监控的 `server` 上，那么建议在此 `vhost` 上禁用统计监控功能。否则会出现“127.0.0.1”，`hostname` 等的域名监控信息。

三、监控数据的查看

安装完 `vts` 模块后，可以通过 `nginx status` 接口进行监控数据的查看，比如：<http://127.0.0.1/status>：

4.

Nginx Vhost Traffic Status

Server main

Host	Version	Uptime	Connections				Requests				Shared memory		
			active	reading	writing	waiting	accepted	handled	Total	Req/s	name	maxSize	used
x1	1.11.10	17h 3m 10s	1	0	1	0	509	509	1587750	4350	vhost_traffic_status	8.0 MiB	8.0 MiB

Server zones

Zone	Requests			Responses						Traffic						
	Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx	Total	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired
server0	8959	0	10ms	0	8959	0	0	0	8959	3.3 MiB	1.6 MiB	0 B	0 B	0	0	0
server1	9585	0	32ms	0	9500	0	85	0	9585	22.8 MiB	1.6 MiB	0 B	0 B	0	0	0
server2	8191	0	5ms	0	8191	0	0	0	8191	512 MiB	32 MiB	0 B	0 B	0	0	2048
server3	41949	0	8ms	0	41949	0	0	0	41949	128 MiB	24 MiB	0 B	0 B	0	0	0
server4	67295	0	10ms	0	67295	0	0	0	67295	512 MiB	64.8 MiB	0 B	0 B	0	0	512
*	135979	0	89ms	0	135894	0	85	0	135979	1178.1 MiB	124 MiB	0 B	0 B	0	0	2560

Upstreams

group0

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Total	Sent
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx		
10.0.0.11:80	up	20ms	10	1	10	0	0	20ms	0	512	0	0	0	512	32 MiB
10.0.0.12:80	up	15ms	10	1	10	0	0	15ms	0	480	0	0	0	480	30 MiB

group1

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Total	Sent
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx		
10.0.0.111:9000	up	80ms	10	1	10	0	0	80ms	0	8192	0	0	0	8192	2560
10.0.0.112:9000	up	15ms	10	1	10	0	0	15ms	0	9110	0	0	0	9110	2560

/tmp/php-fpm.sock

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses				
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx
unix:/tmp/php-fpm.sock	up	25ms	1	1	10	13	0	25ms	0	0	0	0	13

::noproxy

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Responses					Total	Sent
						Total	Req/s	Time	1xx	2xx	3xx	4xx	5xx		
10.0.1.11:8080	up	80ms	10	1	10	0	0	80ms	0	8192	0	0	0	8192	2560
127.0.0.1:8080	up	15ms	10	1	10	0	0	15ms	0	9110	0	0	0	9110	2560

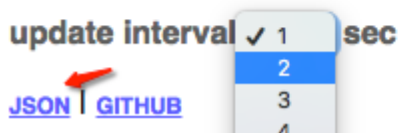
Caches

Zone	Size		Traffic				Cache							
	Capacity	Used	Sent	Rcvd	Sent/s	Rcvd/s	Miss	Bypass	Expired	Stale	Updating	Revalidated	Hit	Size
one	64.0 MiB	16.0 MiB	27.6 MiB	8.8 MiB	0 B	0 B	2	0	8192	0	0	0	1048576	64.0 MiB
two	128.0 MiB	32.0 MiB	64 MiB	16 MiB	0 B	0 B	0	0	512	0	0	0	20170509	128.0 MiB

在页面的最下方可以指定监控页面刷新的时间间隔, 点击 JSON, 可以转为 JSON 格式输出。

nginx_upstream

Server	State	Response Time	Weight	MaxFails	FailTimeout	Requests			Resp		
						Total	Req/s	Time	1xx	2xx	3xx
1:8080	up	0ms	4	100	10	0	0	0ms	0	0	0
1:80	backup	0ms	1	1	10	0	0	0ms	0	0	0



三、nginx-vts-exporter 的使用

exporter 会收集 nginx 性能指标的 JSON 格式数据，并汇总后暴露监控接口给 Prometheus。

它的安装使用很简单，开箱即用：

1. 下载当前最新版本的软件包：

```
# wget -c
```

```
https://github.com/hn1q715/nginx-vts-exporter/releases/download/v0.9.1/nginx-vts-exporter-0.9.1.linux-amd64.tar.gz
```

2. 解压后运行：

```
# nginx-vts-exporter-0.9.1.linux-amd64/nginx-vts-exporter -nginx.scrape_timeout 10 -nginx.scrape_uri
```

```
http://127.0.0.1/status/format/json
```

推荐 exporter 和 nginx 安装在同一台机器上，如果不在同一台主机，把 scrape_uri 改为 nginx 主机的地址。

nginx_vts_exporter 的默认端口号：9913，对外暴露监控接口

```
http://xxx:9913/metrics.
```

四、Nginx 的监控数据类型

nginx-vts-exporter 的数据类型命名空间默认以“nginx”开头，主要有如下 9 个：

HELP 是对监控条目的解释，TYPE 的格式是：监控条目名称+Prometheus 数据类型：

```
# HELP nginx_server_bytes request/response bytes# TYPE
nginx_server_bytes counter# HELP nginx_server_cache cache counter#
TYPE nginx_server_cache counter# HELP nginx_server_connections nginx
connections# TYPE nginx_server_connections gauge# HELP
nginx_server_requestMsec average of request processing times in
milliseconds# TYPE nginx_server_requestMsec gauge# HELP
nginx_server_requests requests counter, 可以区分状态码# TYPE
nginx_server_requests counter# HELP nginx_upstream_bytes
```

```
request/response bytes# TYPE nginx_upstream_bytes counter# HELP
nginx_upstream_requestMsec average of request processing times in
milliseconds# TYPE nginx_upstream_requestMsec gauge# HELP
nginx_upstream_requests requests counter, 可以区分状态码# TYPE
nginx_upstream_requests counter# HELP nginx_upstream_responseMsec
average of only upstream/backend response processing times in milliseconds#
TYPE nginx_upstream_responseMsec gauge
```

五、Nginx 监控在 Prometheus 的数据汇总

常用监控汇总表达式：

DomainName 对应 nginx conf 里的 server_name，这里可以根据不同的 server_name 和 upstream 分别进行 qps、2xx/3xx/4xx/5xx 的状态码监控，另外也可以监控 nginx 每台后端 server 的 qps 和后端接口响应时间。

如果不需要区分 server_name，可以把表达式里的 \$DomainName 改为星号，***** 代表所有；

1. 求 Nginx 的 QPS：

```
sum(irate(nginx_server_requests{code="total",host=~"$DomainName"}[5m]))
```

2. 求 4xx 万分率（5xx 类似，code="5xx")：

```
(sum(irate(nginx_server_requests{code="4xx",host=~"$DomainName"}[5m])) /
sum(irate(nginx_server_requests{code="total",host=~"$DomainName"}[5m]))) * 10000
```

3. 求 upstream 的 QPS(示例求 group1 的 qps):

```
sum(irate(nginx_upstream_requests{code="total",upstream="group1"}[5m]))
```

4. 求 upstream 后端 server 的响应时间（示例求 group1 的后端响应时间）：

```
nginx_upstream_responseMsec{upstream="group1"}
```

六、Nginx 监控的展示

Dashboard 的展示当然是使用 grafana, 自己根据表达式画图即可, 监控图类似:

