

Prometheus进阶

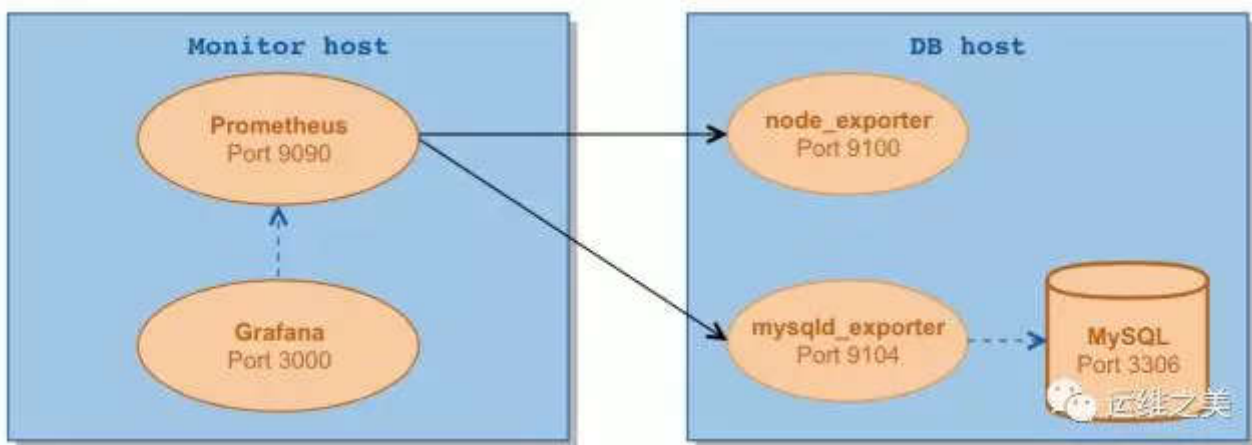
原创 2017-05-24 Mike 运维之美

在「Prometheus入门」一文中我们对Prometheus基本知识点做了讲解，并演示了如何监控一个Linux服务器。这篇文章我们将讲解如何对几个常见的应用进行监控。

监控MySQL服务器

Prometheus通过安装在远程机器上的exporter来收集监控数据，这里要用到的是mysql_exporter。

- 部署的架构图



- 安装mysql_exporter

```
$ wget https://github.com/prometheus/mysql_exporter/releases/download/v0.10.0/mysql
$ tar xzvf mysql_exporter-0.10.0.linux-amd64.tar.gz
$ mv mysql_exporter-0.10.0.linux-amd64 /usr/local/prometheus/mysql_exporter
```

- 增加一个用于监控的MySQL用户

创建一个用于mysql_exporter连接到MySQL的用户并赋予所需的权限。

```
mysql> GRANT REPLICATION CLIENT, PROCESS ON *.* TO 'mysql_exporter'@'localhost' ide
mysql> GRANT SELECT ON performance_schema.* TO 'mysql_exporter'@'localhost';
mysql> flush privileges;
```

- 创建一个用于连接MySQL的配置文件

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

```
$ vim /usr/local/prometheus/mysqld_exporter/.my.cnf

[client]
user=mysqld_exporter
password=000000
```

- 创建Systemd服务

```
$ vim /etc/systemd/system/mysql_exporter.service

[Unit]
Description=mysql_exporter
After=network.target
[Service]
Type=simple
User=prometheus
ExecStart=/usr/local/prometheus/mysqld_exporter/mysqld_exporter -config.my-cnf="/usr
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

- 启动mysqld_exporter

```
$ systemctl start mysql_exporter
```

- 验证mysqld_exporter是否启动成功

```
$ systemctl status mysql_exporter
• mysql_exporter.service - mysql_exporter
   Loaded: loaded (/etc/systemd/system/mysql_exporter.service; disabled; vendor pres
   Active: active (running) since Tue 2017-05-23 14:11:25 CST; 3s ago
   Main PID: 15026 (mysqld_exporter)
     Tasks: 4
    Memory: 1.6M
         CPU: 16ms
    CGroup: /system.slice/mysql_exporter.service
            └─15026 /usr/local/prometheus/mysqld_exporter/mysqld_exporter -config.my-
```

- 修改prometheus.yml，加入下面的监控目标：

mysqld_exporter默认的抓取地址为 `http://IP:9104/metrics`

```
$ vim /usr/local/prometheus/prometheus.yml

- job_name: mysql
  static_configs:
    - targets: ['192.168.2.210:9104']
```

```
labels:
  instance: db1
```

- 重启Prometheus

```
$ systemctl restart prometheus
```

- 在Grafana中导入模板

Grafana目前官方还没有的配置好的MySQL图表模板，这里使用Percona开源的模板。

a) 下载Percona提供的模板

```
$ git clone https://github.com/percona/grafana-dashboards.git
```

Percona提供的模板相当丰富，有MySQL、MariaDB、MongoDB等。

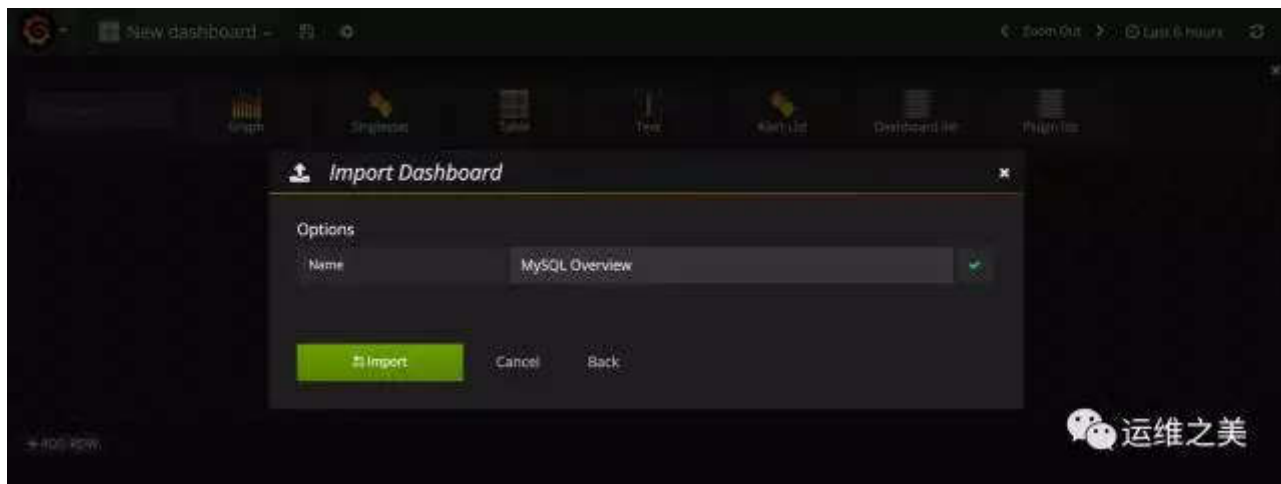
```
ls grafana-dashboards/dashboards/MySQL*

grafana-dashboards/dashboards/MySQL_InnoDB_Metrics_Advanced.json
grafana-dashboards/dashboards/MySQL_Overview.json
grafana-dashboards/dashboards/MySQL_Replication.json
grafana-dashboards/dashboards/MySQL_User_Statistics.json
grafana-dashboards/dashboards/MySQL_InnoDB_Metrics.json
grafana-dashboards/dashboards/MySQL_Performance_Schema.json
grafana-dashboards/dashboards/MySQL_Table_Statistics.json
grafana-dashboards/dashboards/MySQL_MyISAM_Metrics.json
grafana-dashboards/dashboards/MySQL_Query_Response_Time.json
grafana-dashboards/dashboards/MySQL_TokuDB_Metrics.json
```

b) 导入模板

1. 单个导入

以MySQL_Overview模板为例，在Grafana—Dashboard中导入这个文件，数据源选择Prometheus。



2.批量导入

复制所有模板到指定位置

```
$ cp -r grafana-dashboards/dashboards /var/lib/grafana/
```

编辑Grafana配置文件

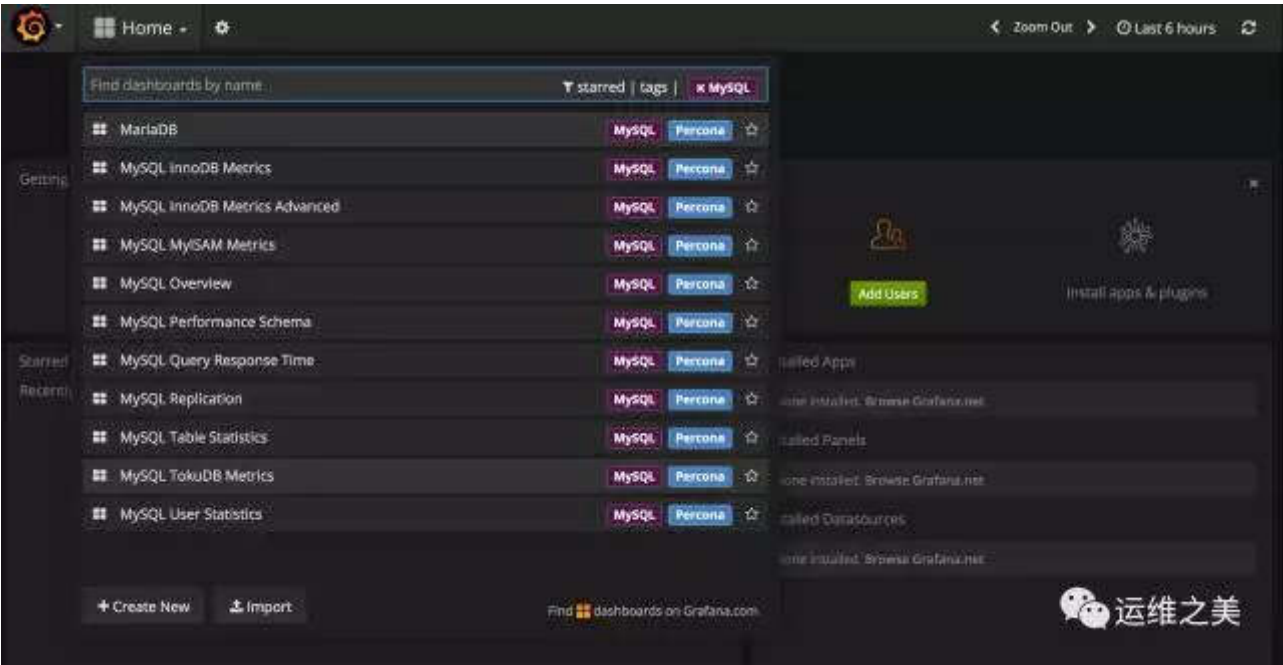
```
$ vim /etc/grafana/grafana.ini

# 修改以下选项
[dashboards.json]
enabled = true
path = /var/lib/grafana/dashboards
```

重启Grafana

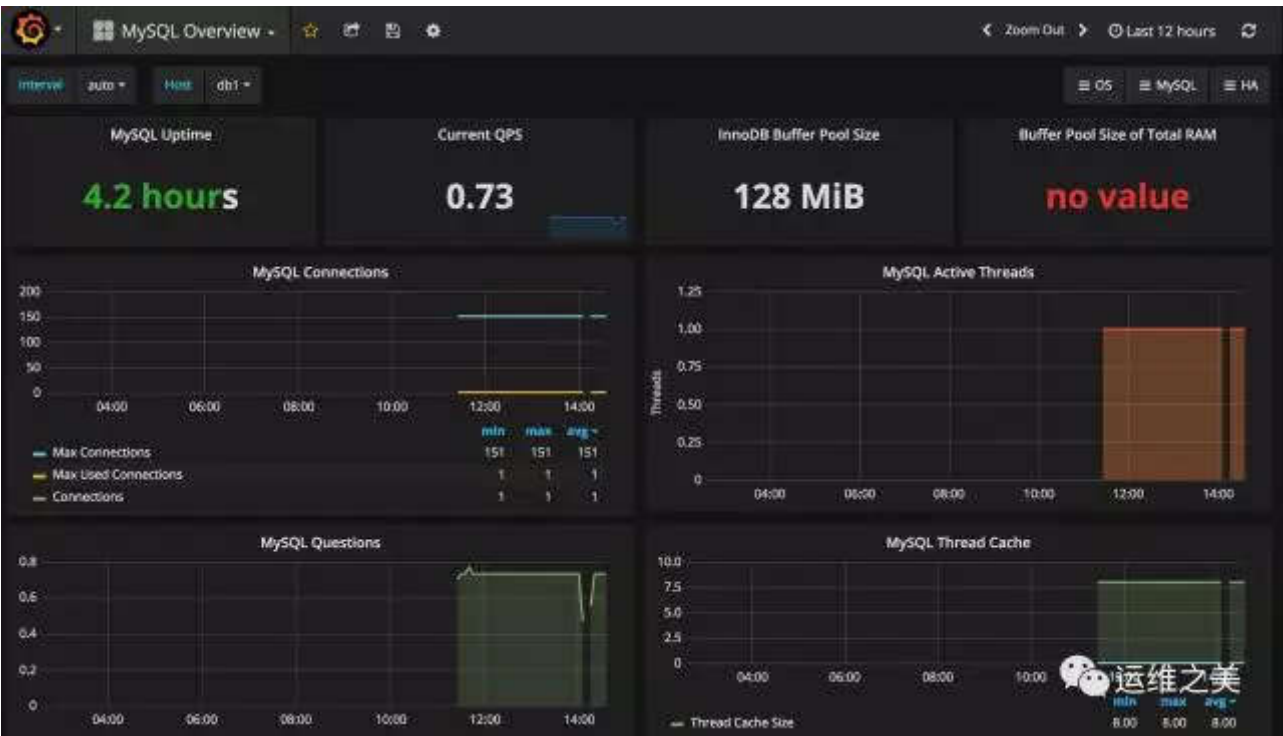
```
$ systemctl restart grafana-server
```

可以看到已批量导入了Percona系列模板。

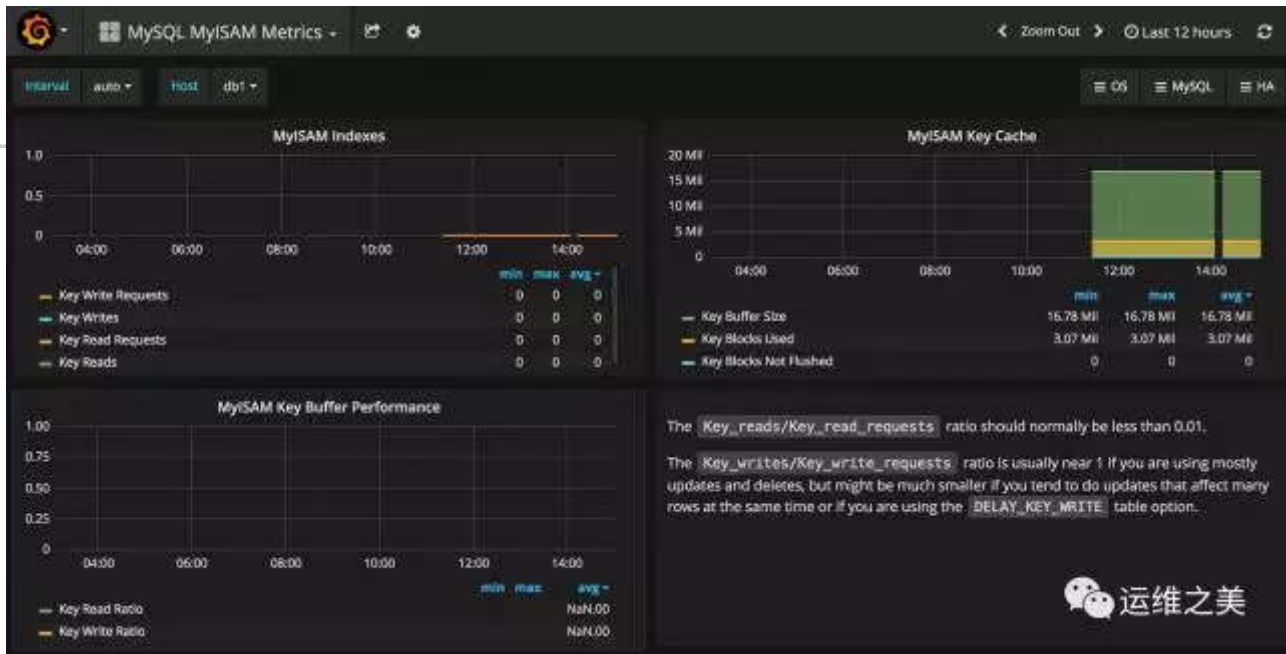


- 访问Dashboards

在Dashboards上选MySQL Overview模板，就可以看到被监控MySQL服务器的各项状态。



其它一些模板的效果



如果你想更加方便的实现MySQL的监控，可以直接使用Percona发布的的监控工具Percona Monitoring and Management(PMM)。具体可以参考「Percona监控工具初探」一文。

监控Nginx服务器

由于官方没有提供Nginx直接可用的exporter，Nginx的监控要相对复杂一些。这里使用的是三方提供nginx-vts-exporter。

- 安装Nginx

由于nginx-vts-exporter依赖于Nginx的nginx-module-vts模块，所以这里需要重新编译下Nginx。

a) 下载对应软件包

```
$ cd /root
$ wget 'http://nginx.org/download/nginx-1.9.2.tar.gz'
$ git clone git://github.com/vozlt/nginx-module-vts.git
```

b) 编译安装Nginx

```
$ apt-get install libreadline-dev libncurses5-dev libpcre3-dev libssl-dev perl make
$ tar xzvf nginx-1.9.2.tar.gz
$ cd nginx-1.9.2
$ ./configure --add-module=/root/nginx-module-vts
$ make && make install
```

c) 修改Nginx配置

这里就不展开讲了，主要需修改内容如下：

```
http {
    vhost_traffic_status_zone;

    ...

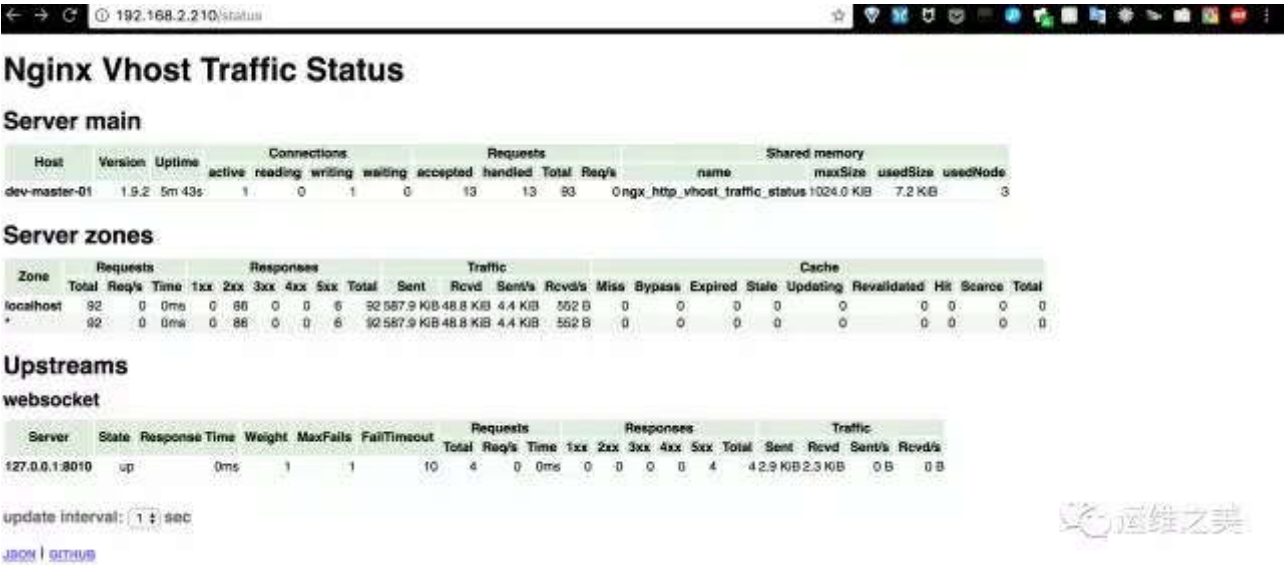
    server {

        ...

        location /status {
            vhost_traffic_status_display;
            vhost_traffic_status_display_format html;
        }
    }
}
```

d) 验证nginx-module-vts模块

访问 `http://IP/status`，出现以下页面：



以JSON格式访问



- 安装nginx-vts-exporter

```
$ wget -O nginx-vts-exporter-0.5.zip https://github.com/hnlq715/nginx-vts-exporter/archive/0.5.zip
$ unzip nginx-vts-exporter-0.5.zip
$ mv nginx-vts-exporter-0.5 /usr/local/prometheus/nginx-vts-exporter
$ chmod +x /usr/local/prometheus/nginx-vts-exporter/bin/nginx-vts-exporter
```

- 创建Systemd服务

```
$ vim /etc/systemd/system/nginx_vts_exporter.service
```

```
[Unit]
Description=nginx_exporter
After=network.target

[Service]
Type=simple
User=prometheus
ExecStart=/usr/local/prometheus/nginx-vts-exporter/bin/nginx-vts-exporter -nginx.scr
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

- 启动nginx-vts-exporter

```
$ systemctl start nginx_vts_exporter.service
```

- 验证nginx-vts-exporter是否启动成功

```
$ systemctl status nginx_vts_exporter.service
```

- nginx_vts_exporter.service - nginx_exporter
Loaded: loaded (/etc/systemd/system/nginx_vts_exporter.service; disabled; vendor


```
Active: active (running) since Wed 2017-05-24 10:37:09 CST; 8s ago
Main PID: 5748 (nginx-vts-expor)
Tasks: 4
Memory: 5.5M
CPU: 13ms
CGroup: /system.slice/nginx_vts_exporter.service
└─5748 /usr/local/prometheus/nginx-vts-exporter/bin/nginx-vts-exporter -n
```

- 修改prometheus.yml，加入下面的监控目标：

nginx-vts-exporter默认的抓取地址为 `http://IP:9913/metrics`

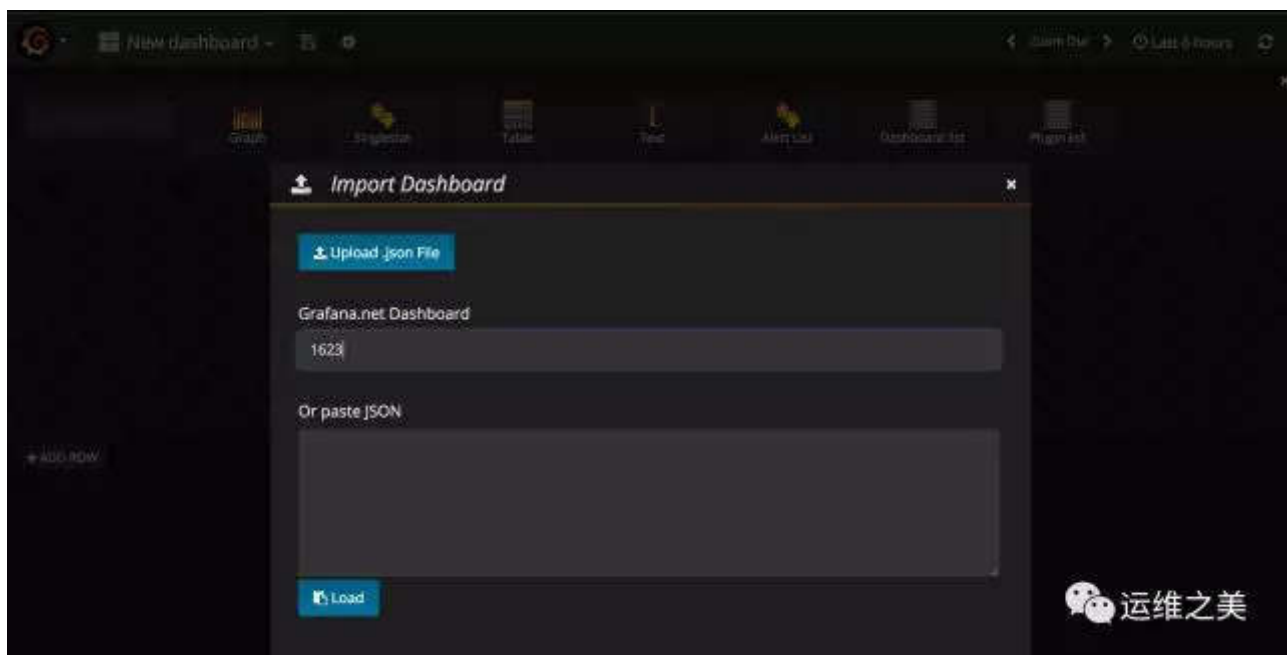
```
$ vim /usr/local/prometheus/prometheus.yml
- job_name: nginx
  static_configs:
    - targets: ['192.168.2.210:9913']
      labels:
        instance: web1
```

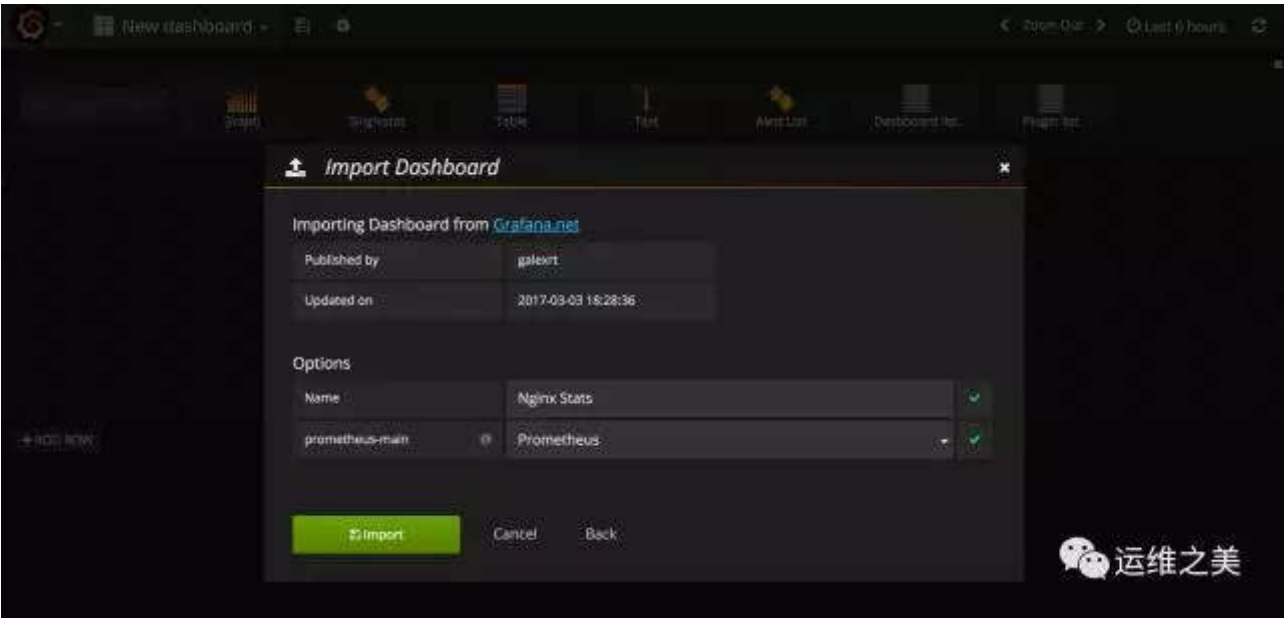
- 重启Prometheus

```
$ systemctl restart prometheus
```

- 导入Nginx Stats模板

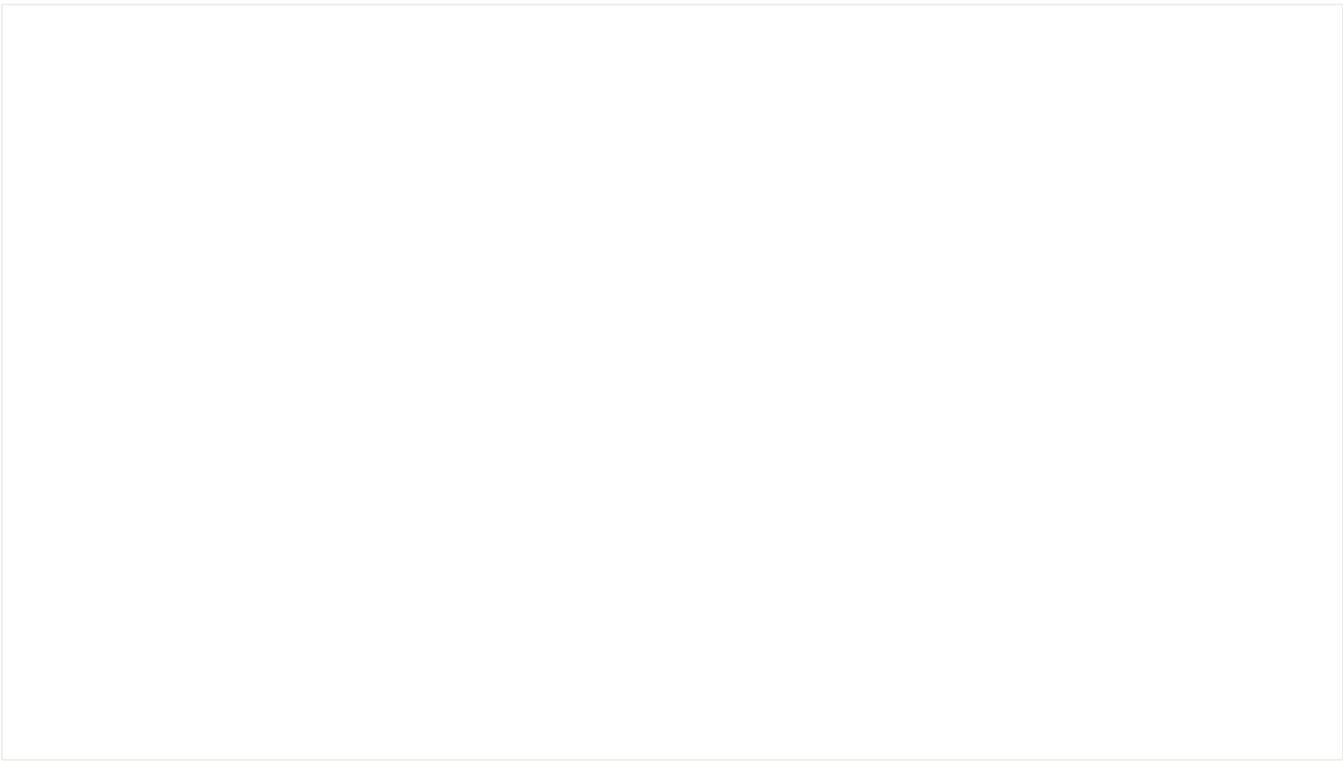
由于是官方平台提供的模板，直接在导入页面填入模板id即可导入。





- 访问Dashboards

在Dashboards上选Nginx Stats模板，就可以看到被监控Nginx服务器的各项状态。



不知道是模板问题，还是打开姿势不对。我这里没有出现数据，不过在Prometheus自带的WEB是可以查询到相应监控指标的。


```
$ systemctl status memcached_exporter
● memcached_exporter.service - memcached_exporter
   Loaded: loaded (/etc/systemd/system/memcached_exporter.service; disabled; vendor
   Active: active (running) since Wed 2017-05-24 11:12:08 CST; 26s ago
 Main PID: 7136 (memcached_expor)
    Tasks: 4
   Memory: 808.0K
      CPU: 6ms
   CGroup: /system.slice/memcached_exporter.service
           └─7136 /usr/local/prometheus/memcached_exporter/memcached_exporter --memc
```

- 修改prometheus.yml，加入下面的监控目标：

memcached_exporter默认的抓取地址为 `http://IP:9150/metrics`

```
$ vim /usr/local/prometheus/prometheus.yml

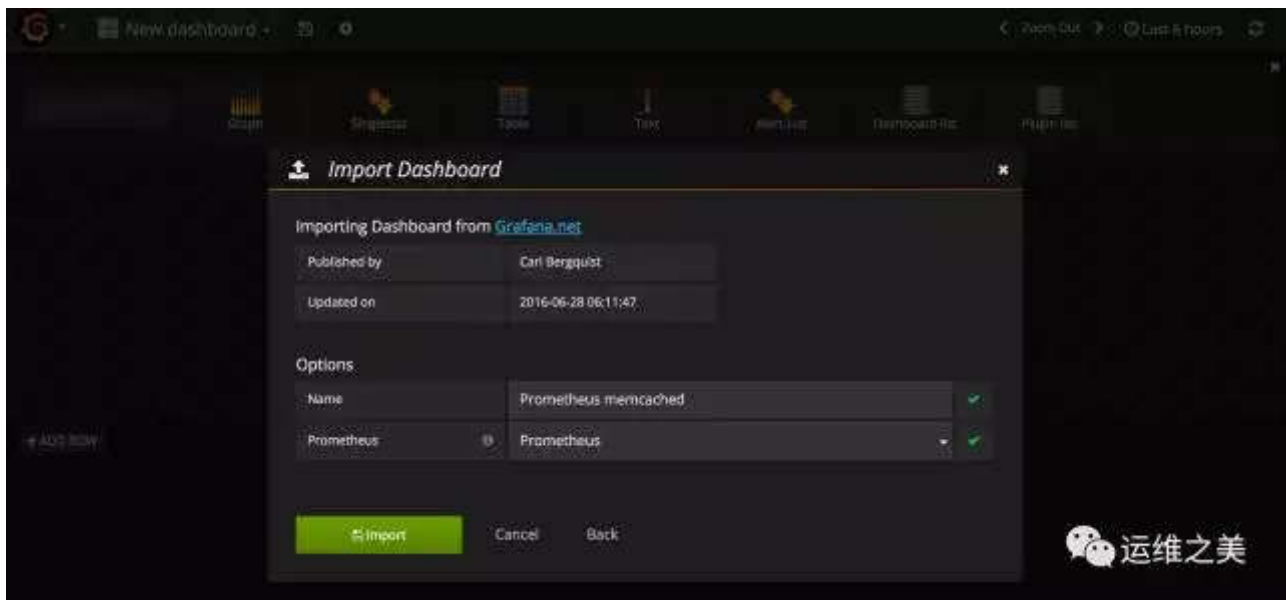
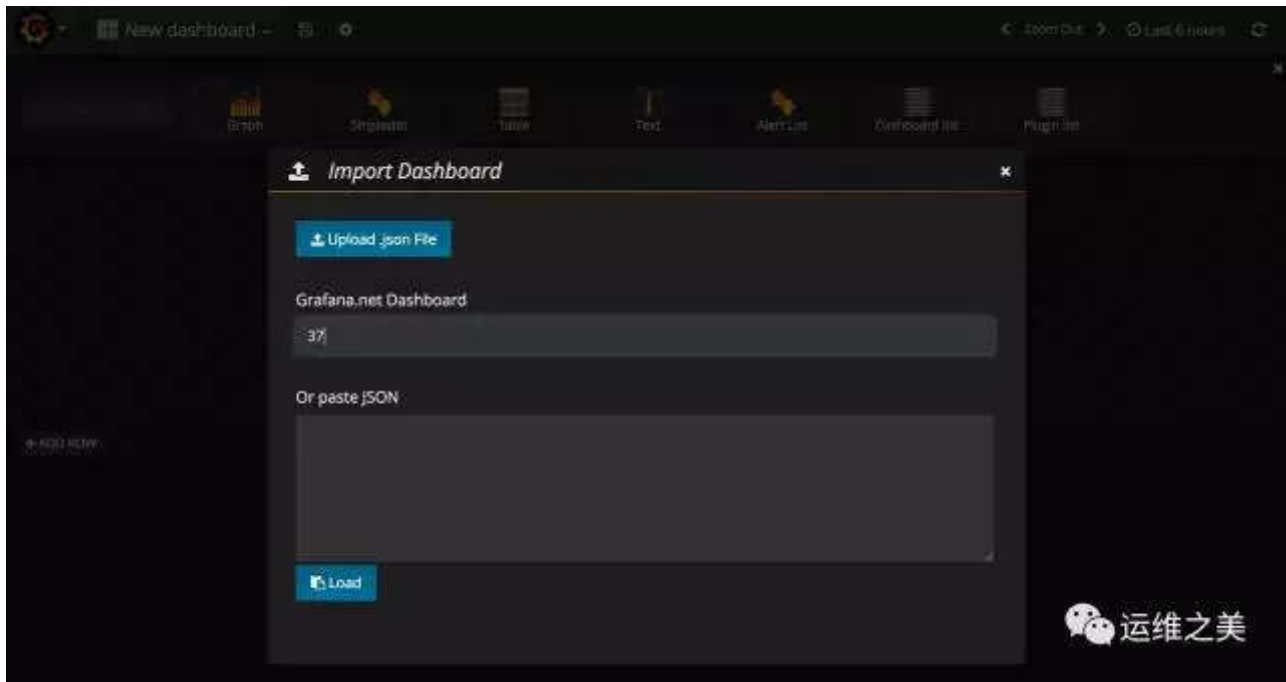
- job_name: memcached
  static_configs:
    - targets: ['192.168.2.210:9150']
      labels:
        instance: db2
```

- 重启Prometheus

```
$ systemctl restart prometheus
```

- 导入Prometheus memcached模板

由于是官方平台提供的模板，直接在导入页面填入模板id即可导入。



- 访问Dashboards

在Dashboards上选Prometheus memcached模板，就可以看到被监控Memcached服务器的各项状态。



好了，这次就先讲几个较常用的监控实例。更多的第三方exporters可参考这里：EXPORTERS AND INTEGRATIONS，目前Grafana官方支持Prometheus的模板还是比较少的。

如果你知道文中Grafana Nginx模板无数据的原因，欢迎留言交流！

参考文档

<http://www.google.com>

<http://qingkang.me/Grafana-Prometheus-Monitor.html>

<https://github.com/percona/grafana-dashboards>

<https://www.percona.com/blog/2016/02/29/graphing-mysql-performance-with-prometheus-and-grafana/>



更多精彩热文：

- Prometheus入门
- Consul入门
- Consul集群部署
- 配置Nginx反向代理WebSocket
- 基于Upsync模块实现Nginx动态配置

运维之美

|

一个有情怀的公众号



长按指纹，识别二维码，加关注



[阅读原文](#)