Grafana是一个开源的度量分析与可视化套件。经常被用作基础设施的时间序列数据和应用程序分析的可视化，它在其他领域也被广泛的使用包括工业传感器、家庭自动化、天气和过程控制等。

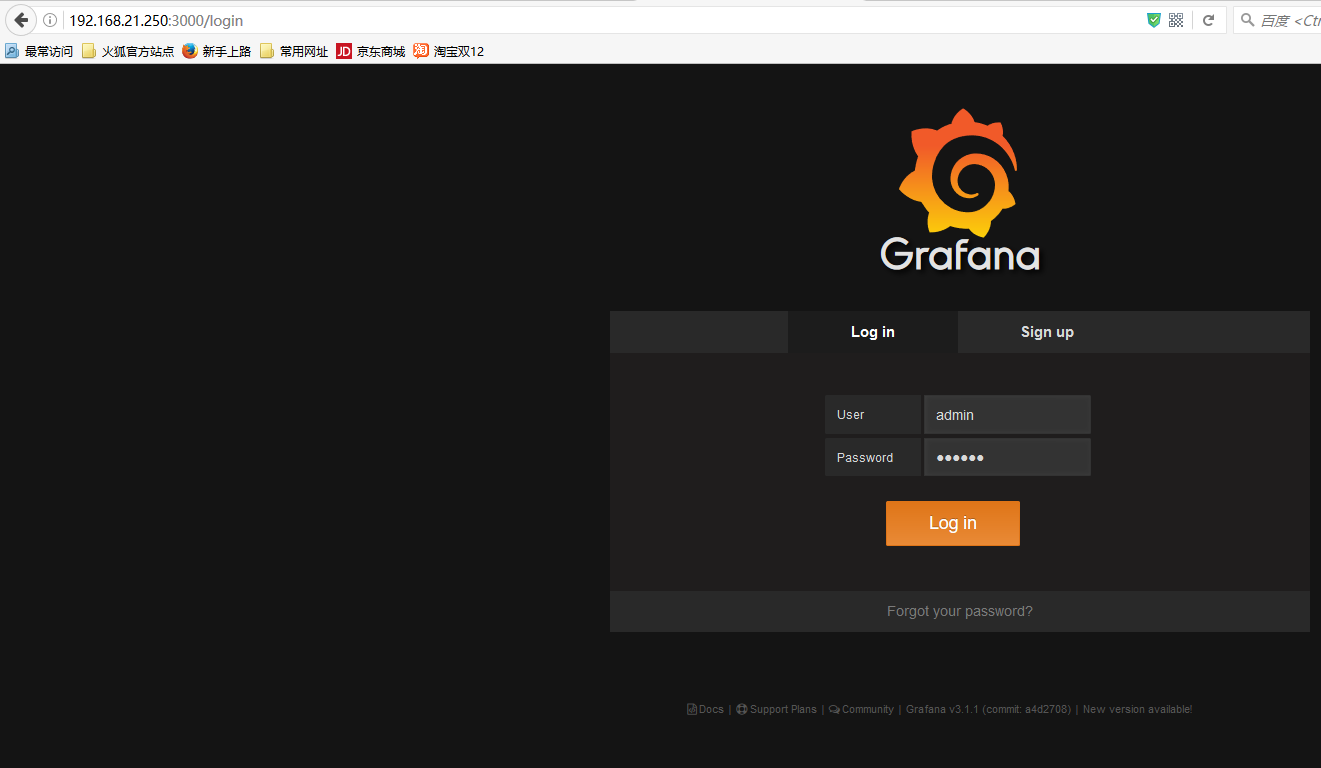
Grafana支持许多不同的数据源。每个数据源都有一个特定的查询编辑器,该编辑器定制的特性和功能是公开的特定数据来源。

 官方支持以下数据源:Graphite，InfluxDB，OpenTSDB，Prometheus，Elasticsearch，CloudWatch和KairosDB。

每个数据源的查询语言和能力都是不同的。你可以把来自多个数据源的数据组合到一个仪表板，但每一个面板被绑定到一个特定的数据源,它就属于一个特定的组织。下面我们来一起感受一下grafana。

**1、登录grafana 界面**

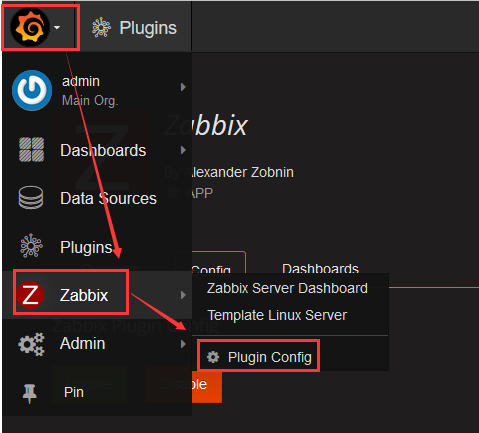
    http://192.168.21.250:3000 默认用户和密码均为**admin**

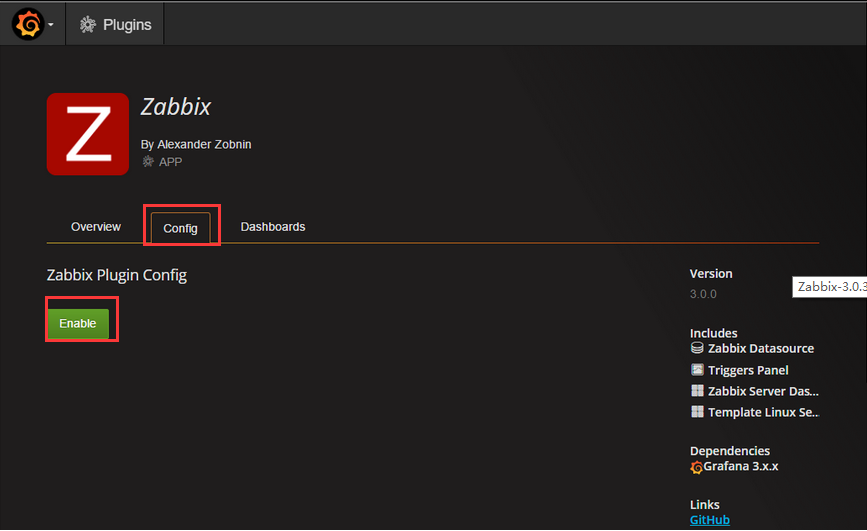


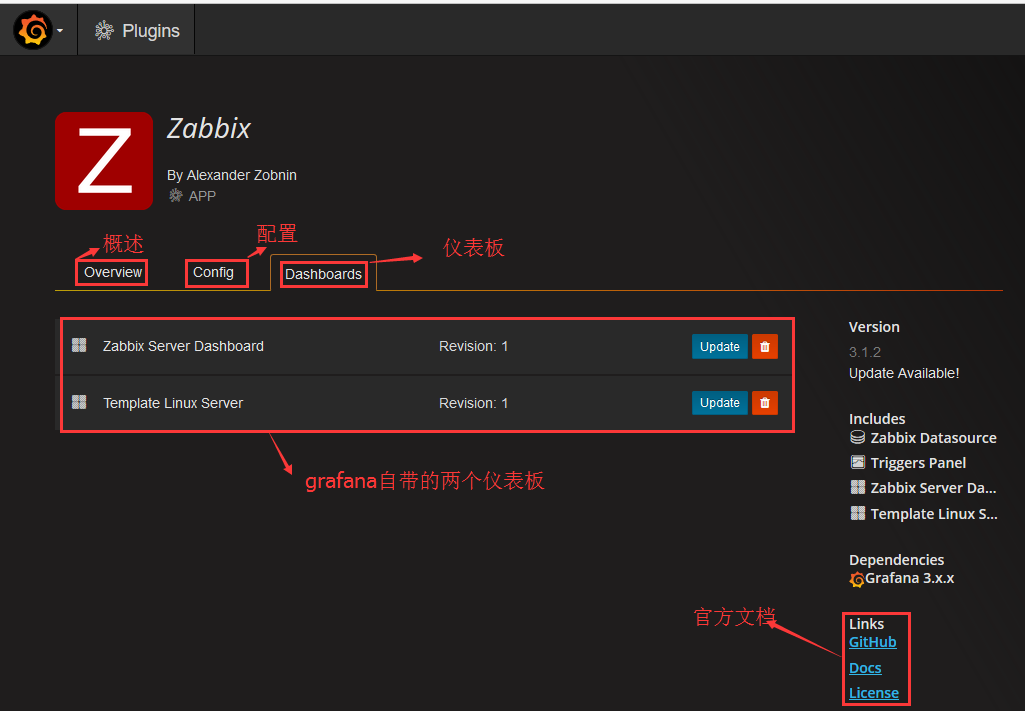
**2、配置Grafana启用zabbix插件**

    登录到grafana,移动鼠标到grafana左上角，单击选择齿轮图标的向下按钮，然后选择“**Zabbix**”选项卡，选择“**Pluugin Config**”,打开Zabbix 插件配置对话框，选择启用zabbix插件。

    具体操作步骤如下图所示：



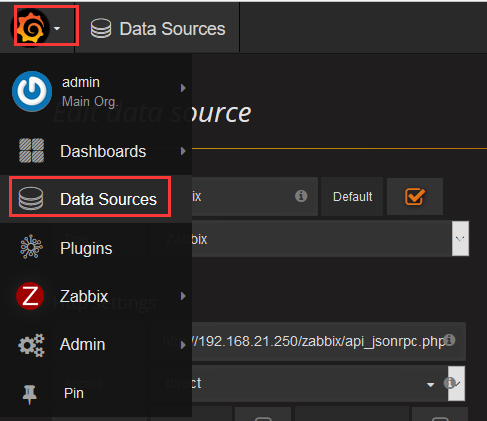




**3、配置Zabbix数据源**

    添加新数据源，单击选择齿轮图标的向下按钮，打开 “**Data Sources”**

，单击“**Add new**”。



注意红线标注的地方

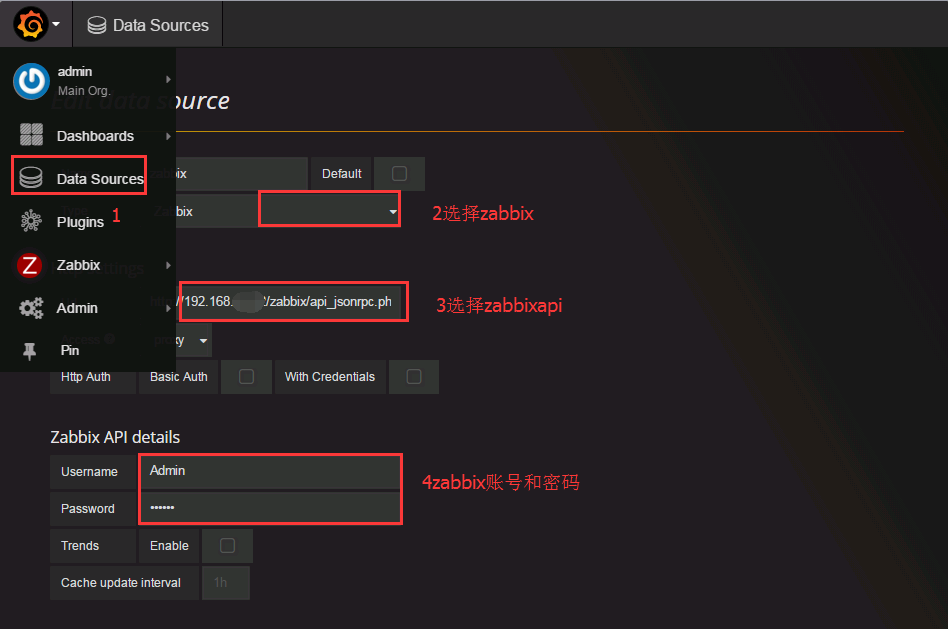
**Name**自定义

**Type**选择**Zabbix**

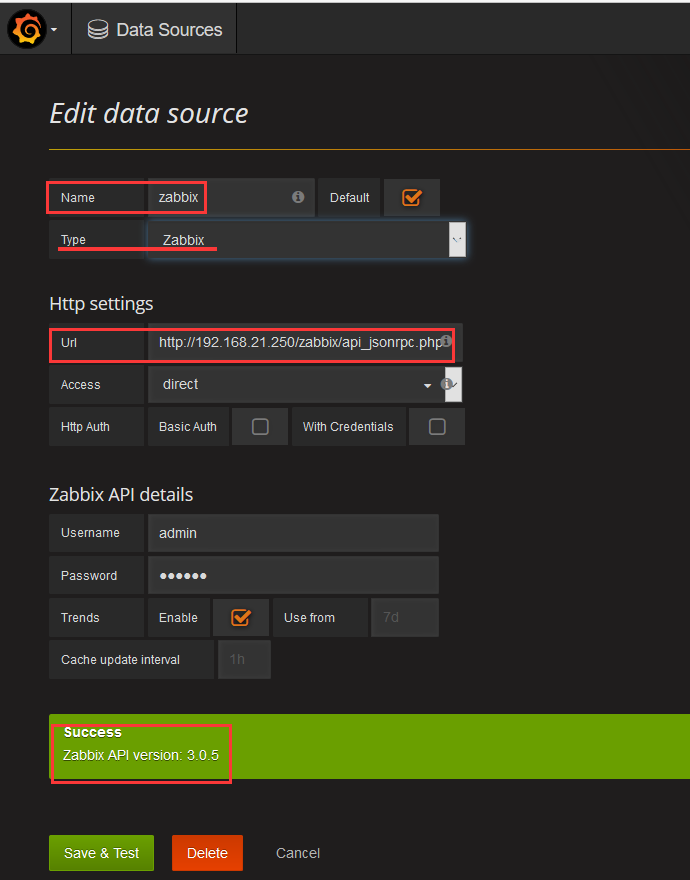
**Http settings URL**填入http://zabbix服务器ip/zabbix/api\_jsonrpc.php

**Zabbix details**用户名和密码需要在Zabbix web页面中设置，本文中用户名：**admin**，密码：**zabbix。**如不想新建的话，可以使用**zabbix**的初始用户。

    配置过程如下图所示：



    设置完成点击**Save & Test**按钮，弹出下图所示的**Success**提示对话框：



**本文档的Zabbix版本为Zabbix-3.0.5，详细配置教程请参考官方文档：**

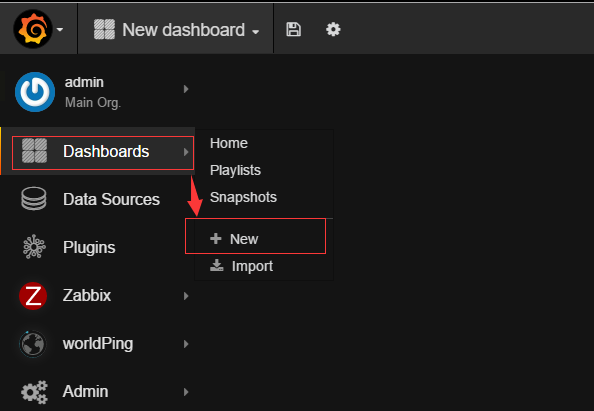
**http://docs.grafana-zabbix.org/installation/configuration**

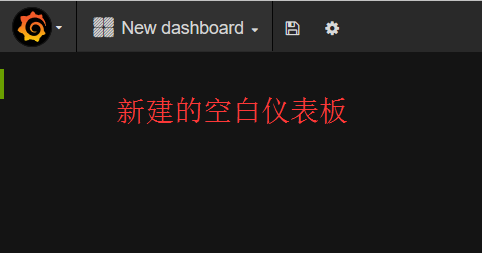
**常见错误解决请参考：http://docs.grafana.org/installation/troubleshooting/**

4、**开始使用Grafana-Zabbix**

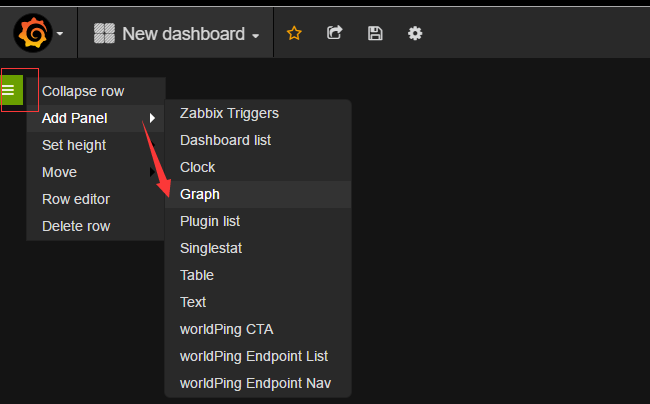
**添加新的仪表板**

    让我们开始创建一个新的仪表板。添加新的仪表板过程如下所示：

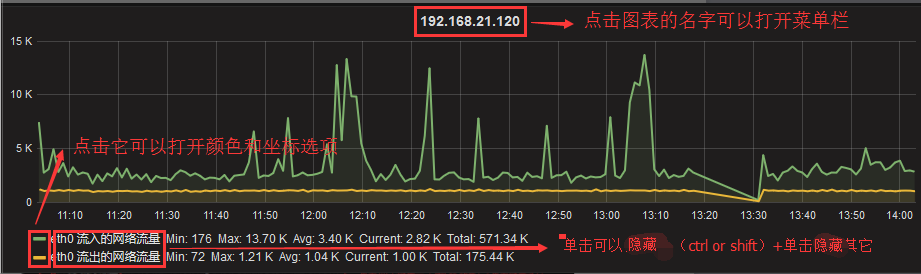
****

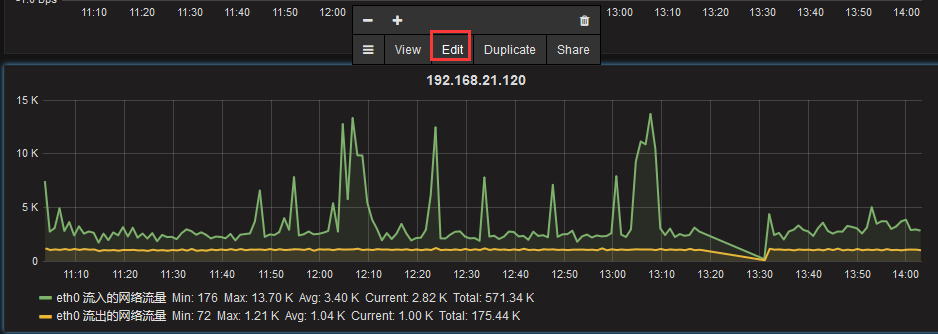


**在新建的仪表板中添加图面板**

****

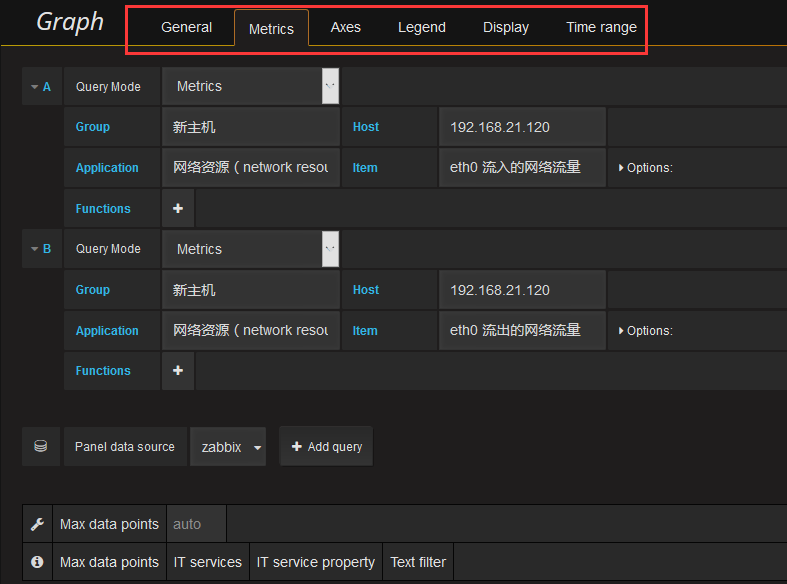
    图面板在Grafana中只是命名图。它提供了一组丰富的图形选项。如下图所示：





    单击标题面板可打开一个菜单框。单击**edit**选项面板将会打开额外的配置选项。

如下图所示：



**Graph里面的选项有：**

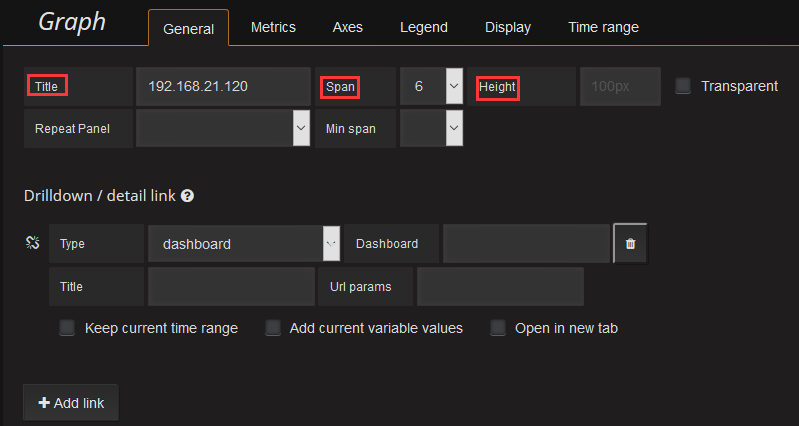
    General（常规选择）、Metrics（指标）、Axes（坐标轴）、Legend（图例）、 Display（显示样式）、Time range（时间范围）

**Genera（常规选择）：添加图形标题，图形宽度高度等**

    Title：仪表板上的面板标题

    Span：列在面板中的宽度

    Height：面板内容高度(以像素为单位)

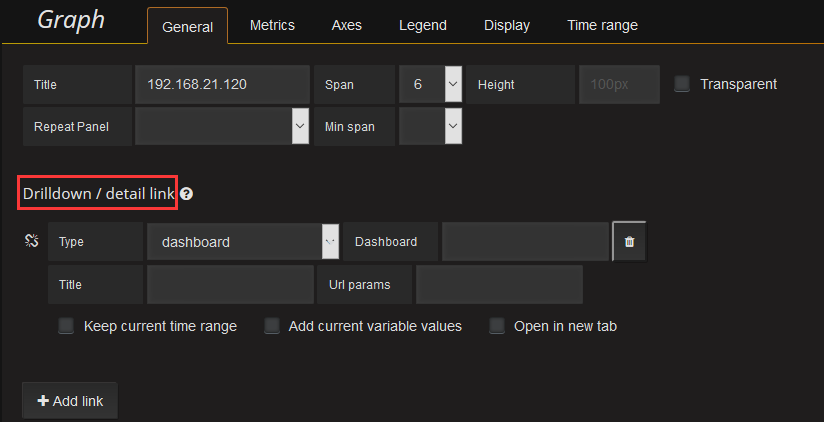


**钻取/详细信息链接（Drilldown / detail link）**

    钻取部分允许添加动态面板的链接，可以链接到其他仪表板或URL。

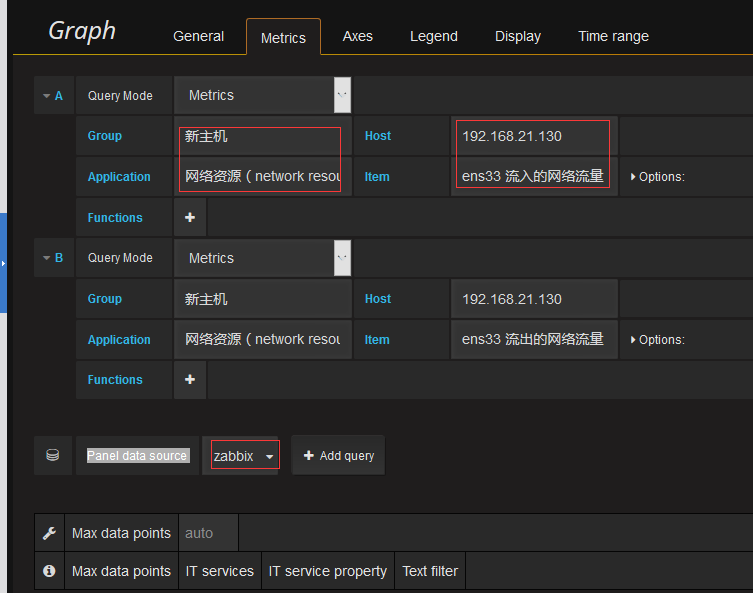
    每个链接都有一个标题,一个类型和参数。链接可以是 dashboard或 absolute链接。如果它是一个仪表板链接, dashboard值必须是一个仪表板的名字。如果这是一个 absolute链接,是URL链接的URL。

**params**允许添加额外的URL参数的链接。格式是 name=value与多个参数分开，当链接到另一个仪表板使用模板变量,你可以使用 var-myvar=value填充模板变量的期望值链接。



**Metrics（指标）**

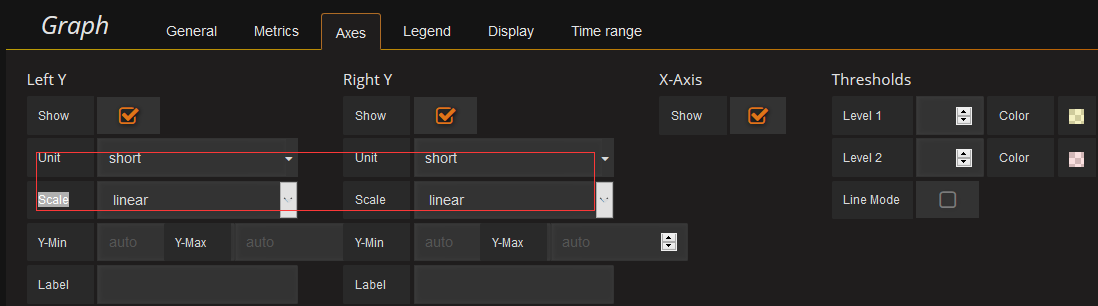
    定义了来源数据的呈现，每个数据源都提供不同的选择。面板的来源数据通过group,host,application,item从zabbix中获得。



**Axes（坐标轴）**

    用于坐标轴和网格的显示方式，包括单位，比例，标签等。

Left Y和 Right Y可以定制使用，因其中的可选参数太多，怕描述不准确。所以请在使用的时候参考官方文档



**Legend（图例）：图例展示**

    图例的参数:

**Total:**返回所有度量查询值的总和

**Current:**返回度量查询的最后一个值

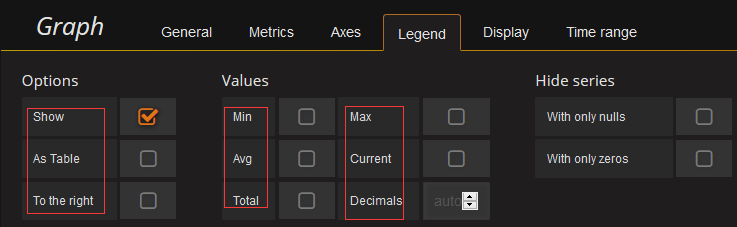
**Min:**返回最小的度量查询值

**Max:**返回最大的度量查询值

**Avg:**返回所有度量查询的平均值

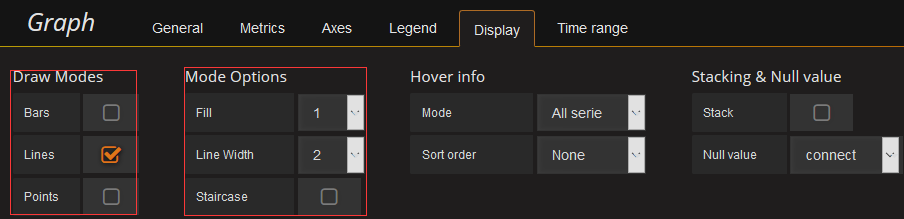
**Decimals:**控制Legend值的多少，以小数显示悬浮工具提示(图)

    Grafana 中Legend值的计算取决于你使用的度量查询方式和什么样类型的聚合或合并点来实现的，所有上述所说的值在同一时间可能都是不正确的。例如，如果你是每秒请求一次,这可能是使用平均值来作为一个整合,然而这个Legend值不会代表请求的总数。这只是Grafana收到的所有数据点的总和。



**Display（显示样式）**

    显示样式的控件属性图如下：



**图表模式(Draw Modes)**

**Bar:**一个条形图显示值

**Lines:**显示线图值

**Points:**显示点值

**选择模式（Mode Options）**

**Fill:**系列的颜色填充,0是没有。

**Line Width:**线的宽度。

**Staircase:**楼梯状显示。

    如果有多个选择项,它们可以作为一个群体显示。

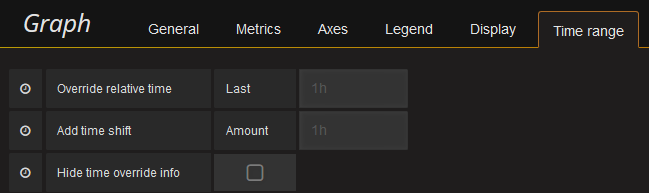
**叠加和空值（Stacking & Null value）**

**Stack：**每个系列是叠在另一个之上

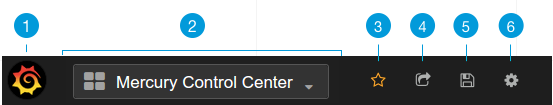
**Null value：**空值

    如果你启用了堆栈可以选择应该显示鼠标悬停功能。

**Time range（时间范围）**

****

**顶级头介绍**

****

上图显示了仪表板顶部的标题。

**1.侧菜单切换:**切换菜单，让你专注于仪表板中给出的数据。侧菜单提供了访问特性，仪表板，用户，组织和数据源等。

**2.仪表板下拉菜单:**下拉菜单显示你当前浏览的仪表板，并允许轻松地切换到另一个新的仪表板。在这里你还可以创建一个新的仪表板，导入现有的仪表板和管理仪表板播放列表。

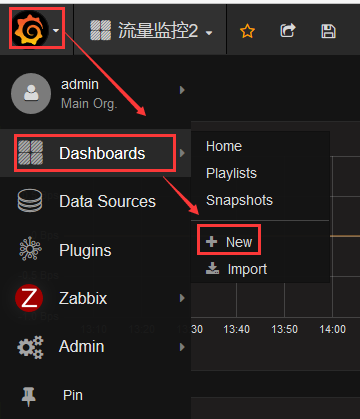
**3.星仪表板:**星(或unstar)当前的仪表板。默认情况下星仪表板将出现在自己建立的仪表板里，为你提供快捷的查看途径。

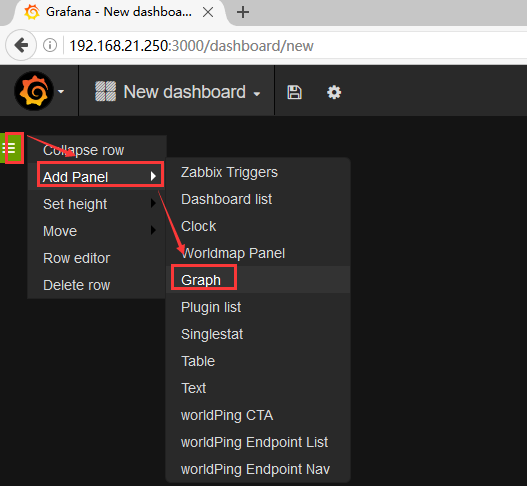
**4.仪表板分享:**通过创建一个链接或创建一个静态快照分享当前仪表板。

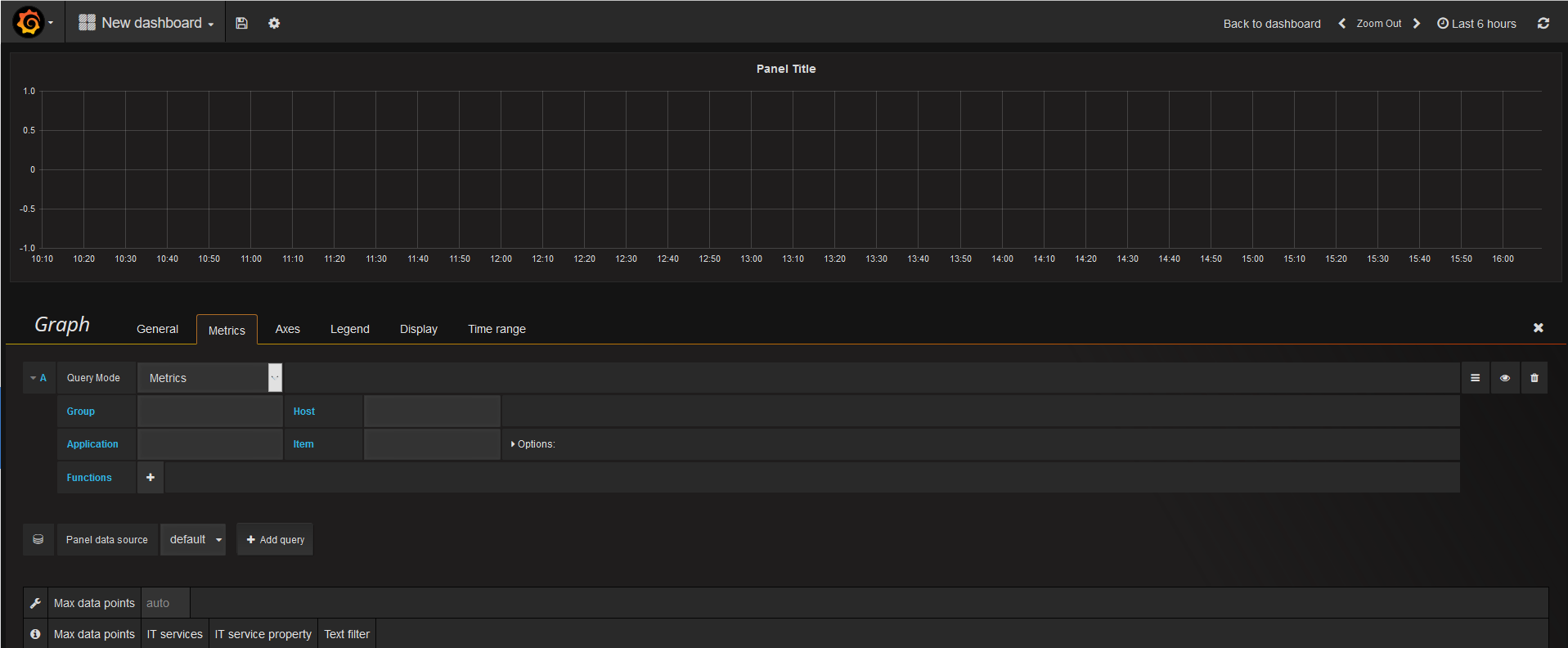
**5.保存仪表板:**以当前仪表板的名字保存。

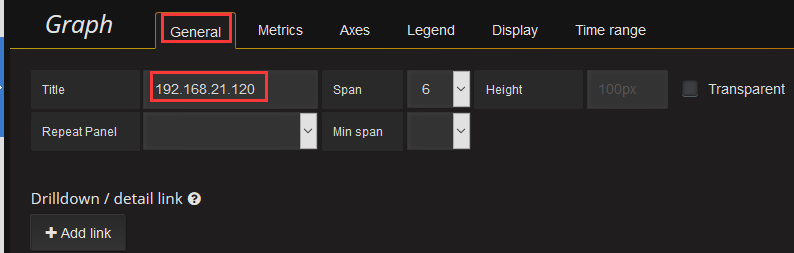
**6.设置:**管理仪表板的设置和特性，比如模板和注释。

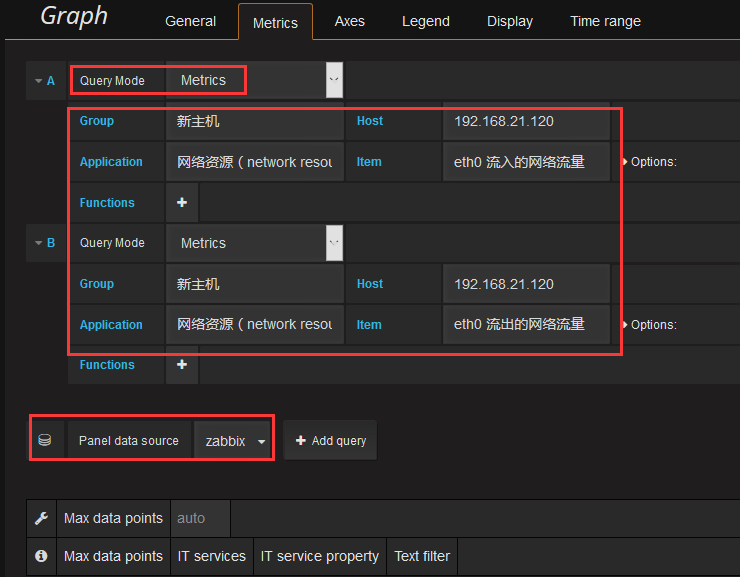
**5、创建流量监控图形**

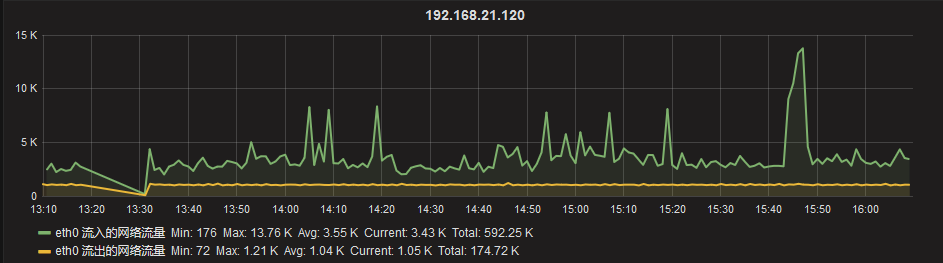
****











**6、仪表盘模板功能**

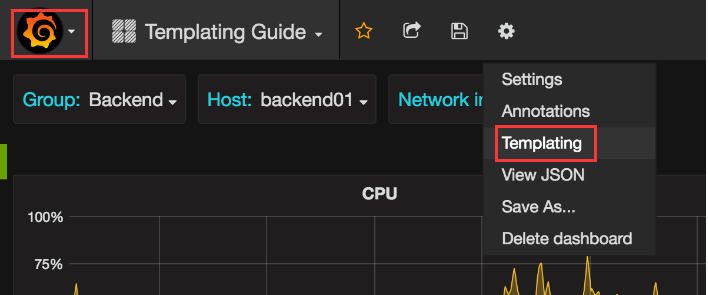
    单纯的手动去添加一个个监控图,只能显示一个主机的所有监控图形，若要查看不同主机的所有监控图形，就要通过变量的方式去实现。我们要设置的变量包括group，host，application和iteam。

**模板**

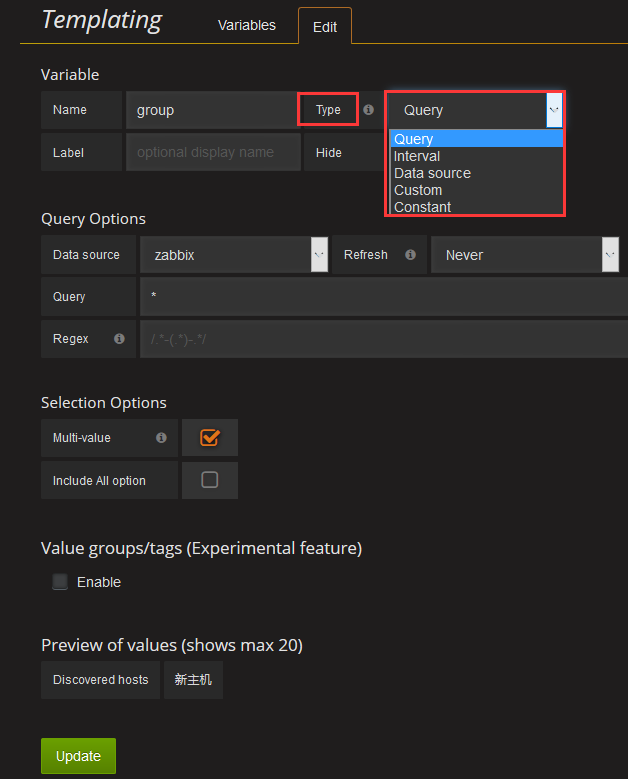
    仪表盘模板可以让你创建一个交互式和动态性的仪表板，它是Grafana里面最强大的、最常用的功能之一。创建的仪表盘模板参数，可以在任何一个仪表盘中使用。

**创建变量**

    点击顶部导航栏上的齿轮图标，选择**模板**。



   单击**新建**按钮，你会看到模板变量编辑器。它包含以下部分：



**变量（Variable）**

**命名：**变量的名称。

**标签：**可见标签变量。例如，主机组，而不是HOST\_GROUP。

**类型：**查询类型选择。

    图中有五种变量类型: query,custom,interval，Data source和Contsta。它们都可以用来创建动态变量,不同之处在于获得的数据值不一样。

**查询选项（Query Options）**

**数据源：**用于查询变量值的数据源。

**刷新：**更新此变量的值。

**查询：**查询字符串。

**正则表达式：**如果你需要筛选值或提取价值的一部分，那就使用正则表达式。

**选择选项（Selection Options）**

**多值：**启用，如果你想在同一时间选择多个值。

**数值组/标签（实验功能）（Value groups/tags (Experimental feature)）**

**7、查询格式**

   zabbix模板变量数据源查询是一个包含了4个部分的以.号隔开的字符串{host group}.{host}.{application}.{item name}。例如， Zabbix servers.Zabbix server.CPU.\*。

**例子：**

**\***返回所有可用主机组

**\*.\***返回主机组里所有可用主机

**Servers.\***返回服务器组里的所有主机

**Linux servers.\*.\*** 返回Linux服务器组中的所有应用程序

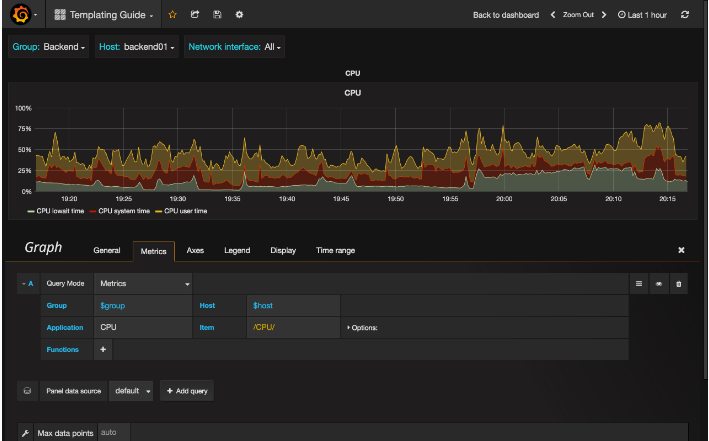
**Linux servers.\*.\*.\***返回Linux服务器组中所有主机的监控项。

   你可以使用另一个变量作为查询的一部分。例如，你有一个变量组，它返回的是主机组的列表，并仅希望将其用于在选定的组在查询主机。下面是这种情况的查询条件：

**$group.\***

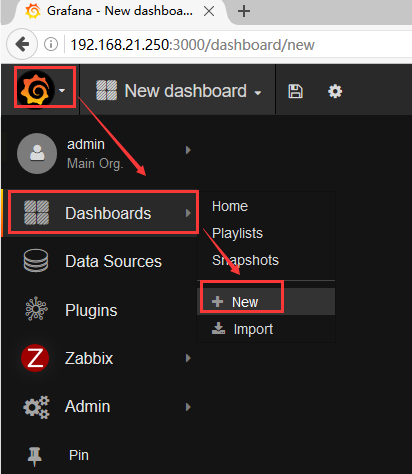
**8、变量的使用**

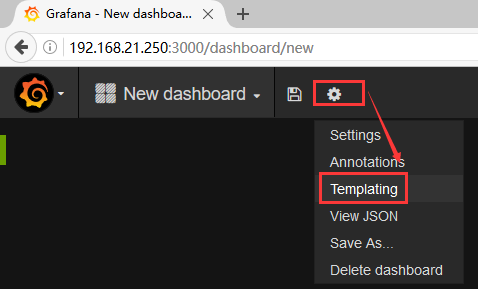
    当你创建一个变量,你可以使用它作为一个数据源查询的一部分。Grafana还支持变量在不同的地方被使用,比如面板和行标题、文本面板的内容等。



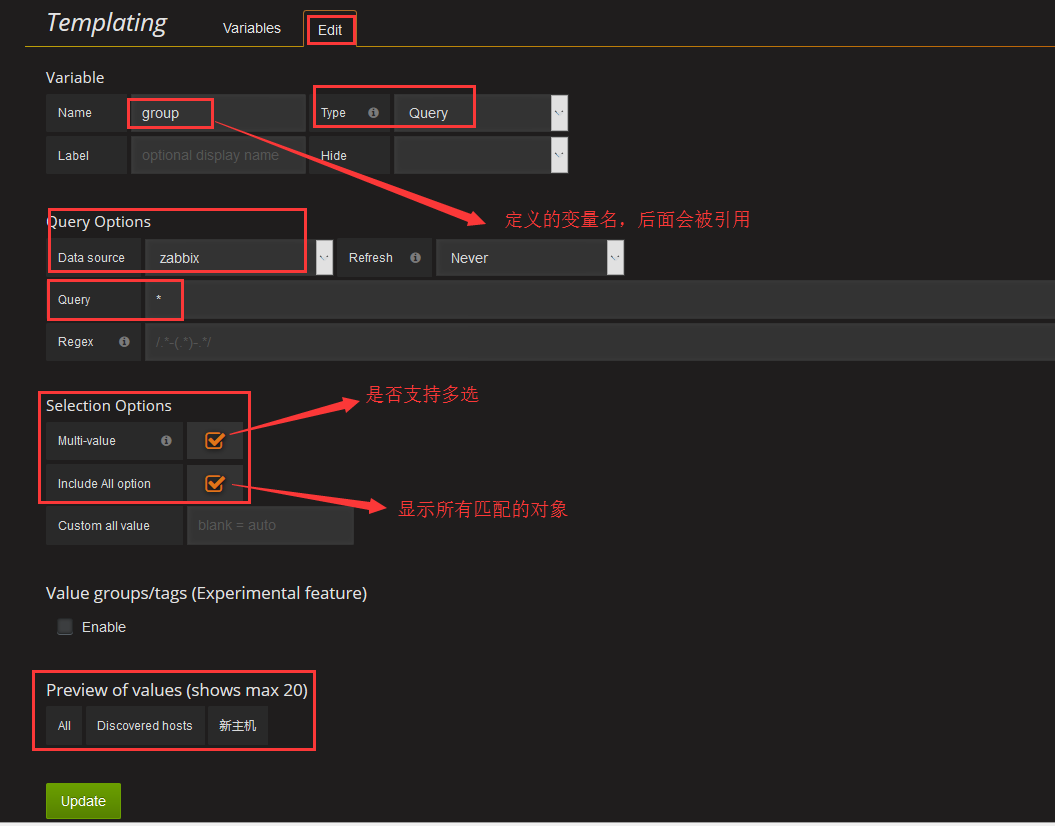
    注意,你需要在变量的名字之前添加$标志。

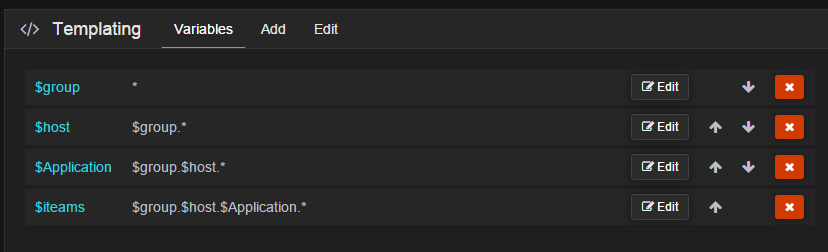
**创建模板**



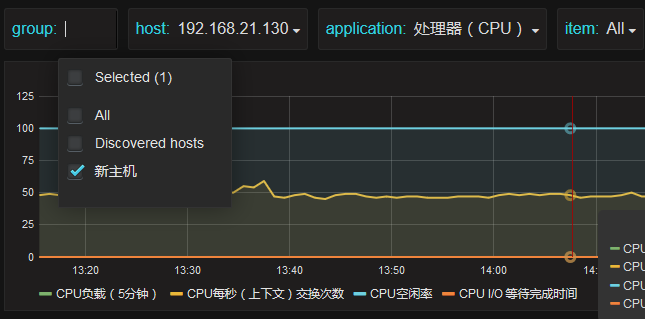


   添加变量group，host，Application，iteam

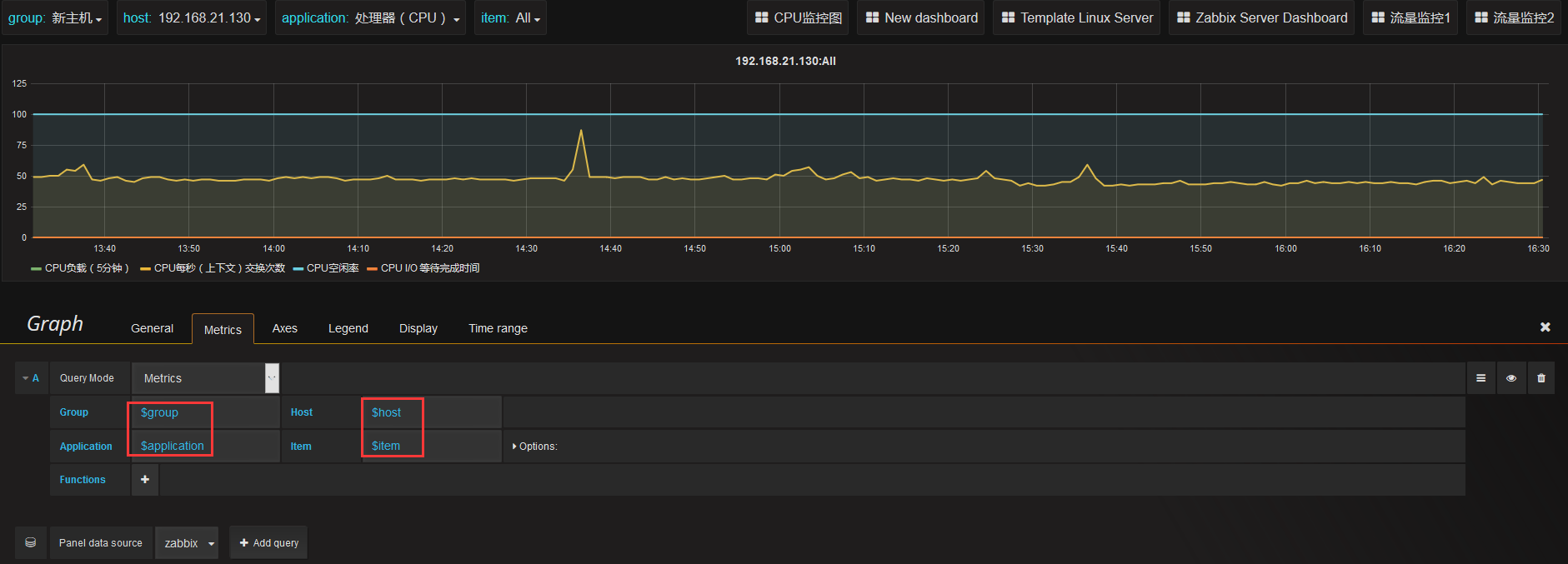
  
    添加完四个变量，如下图所示：



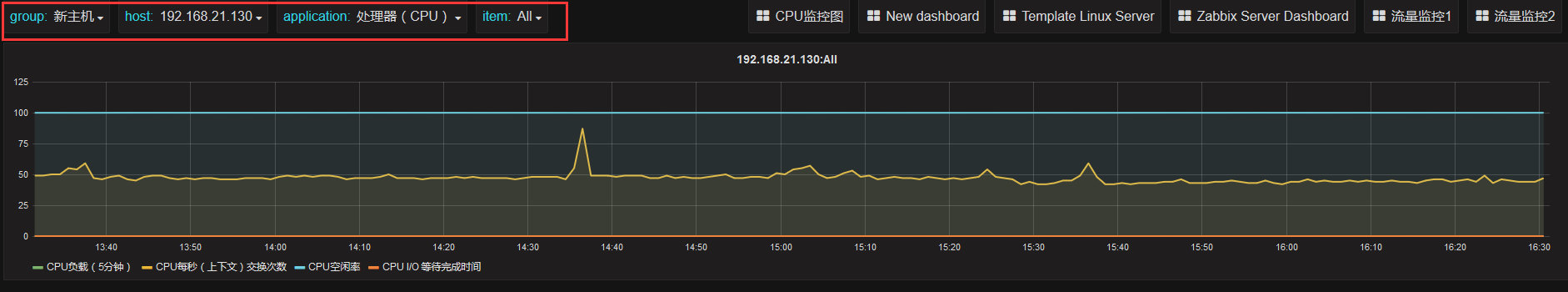
    group匹配的显示结果



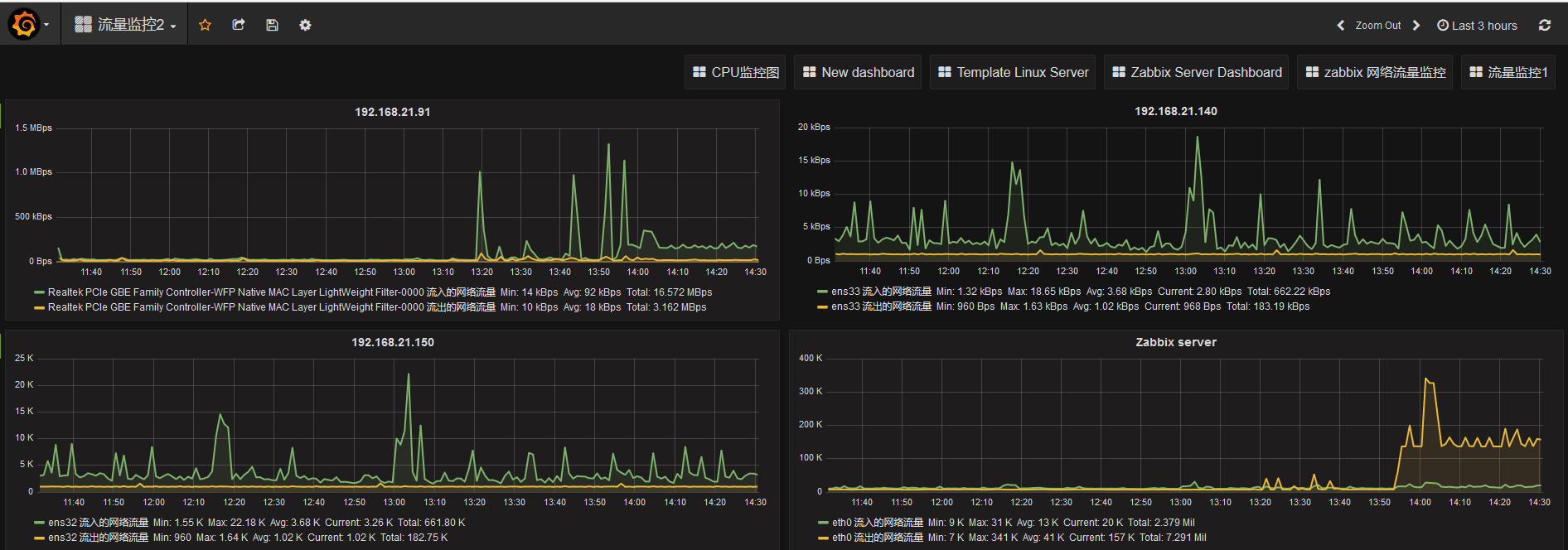
   变量添加完成后，就可以设置图形属性了。将之前所有添加的图形用下面的group，host，application，iteam变量来表示。



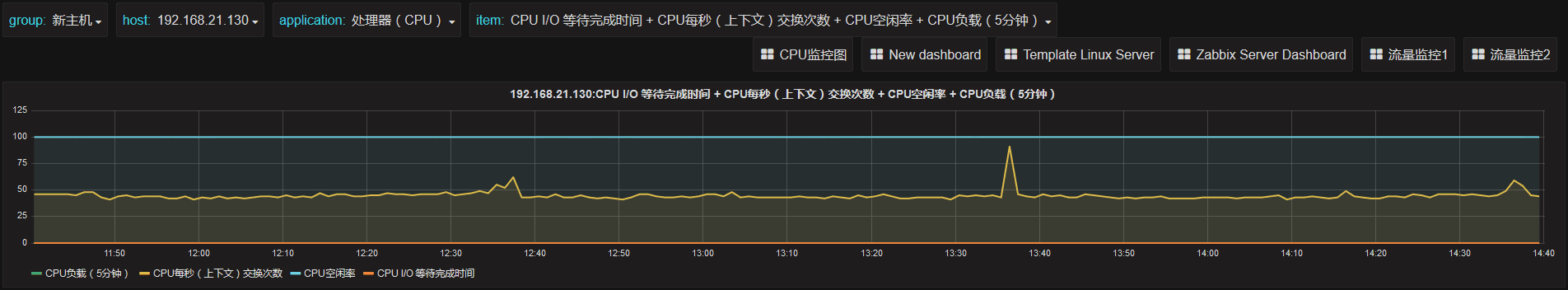
    这样我们就可以通过切换，来查看不同主机的所有监控内容

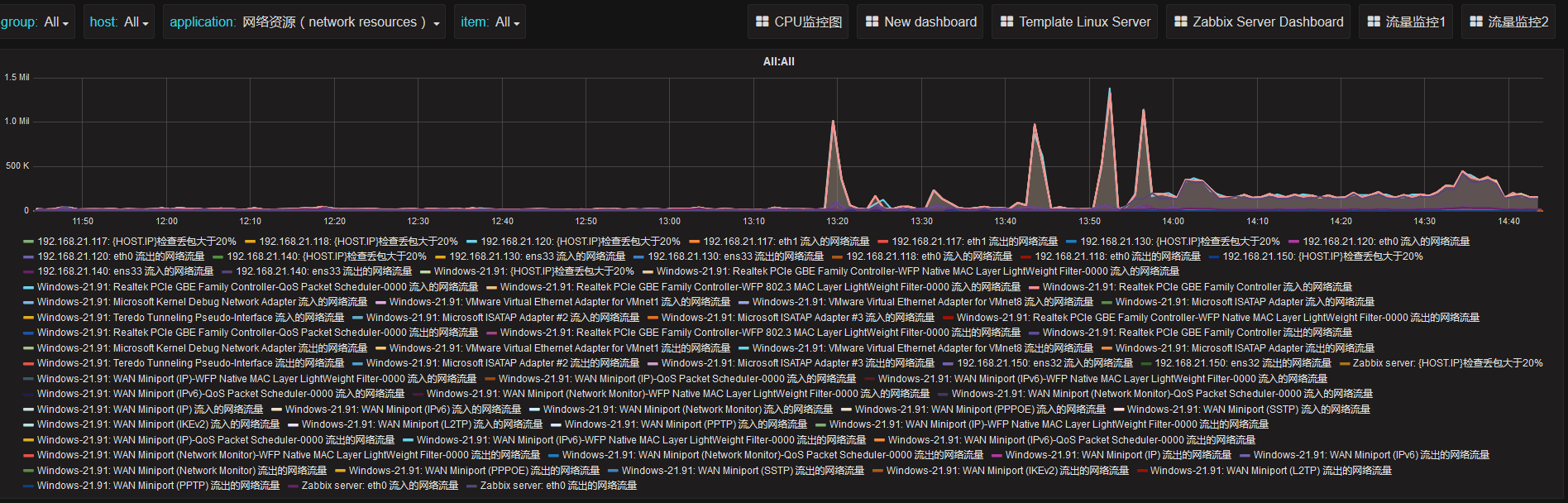


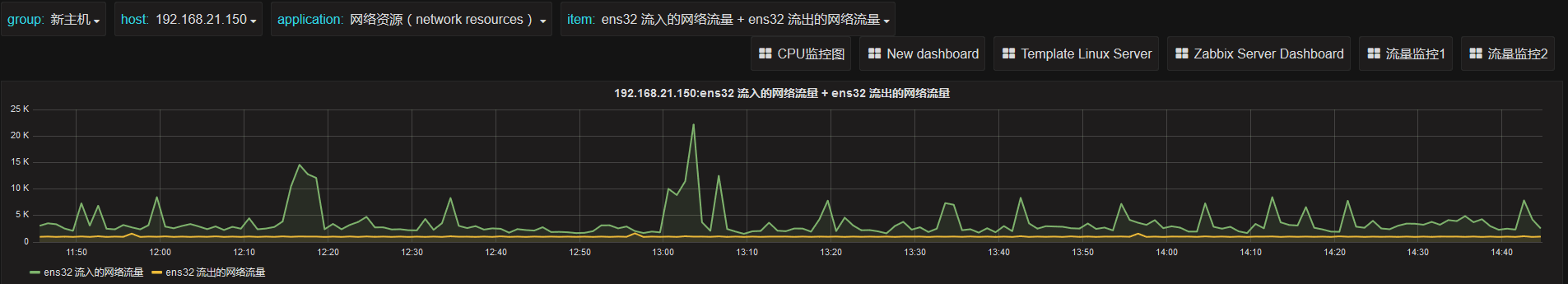
    下图是通过仪表盘功能在一个页面中用多个graph显示多台机器的网卡流量。



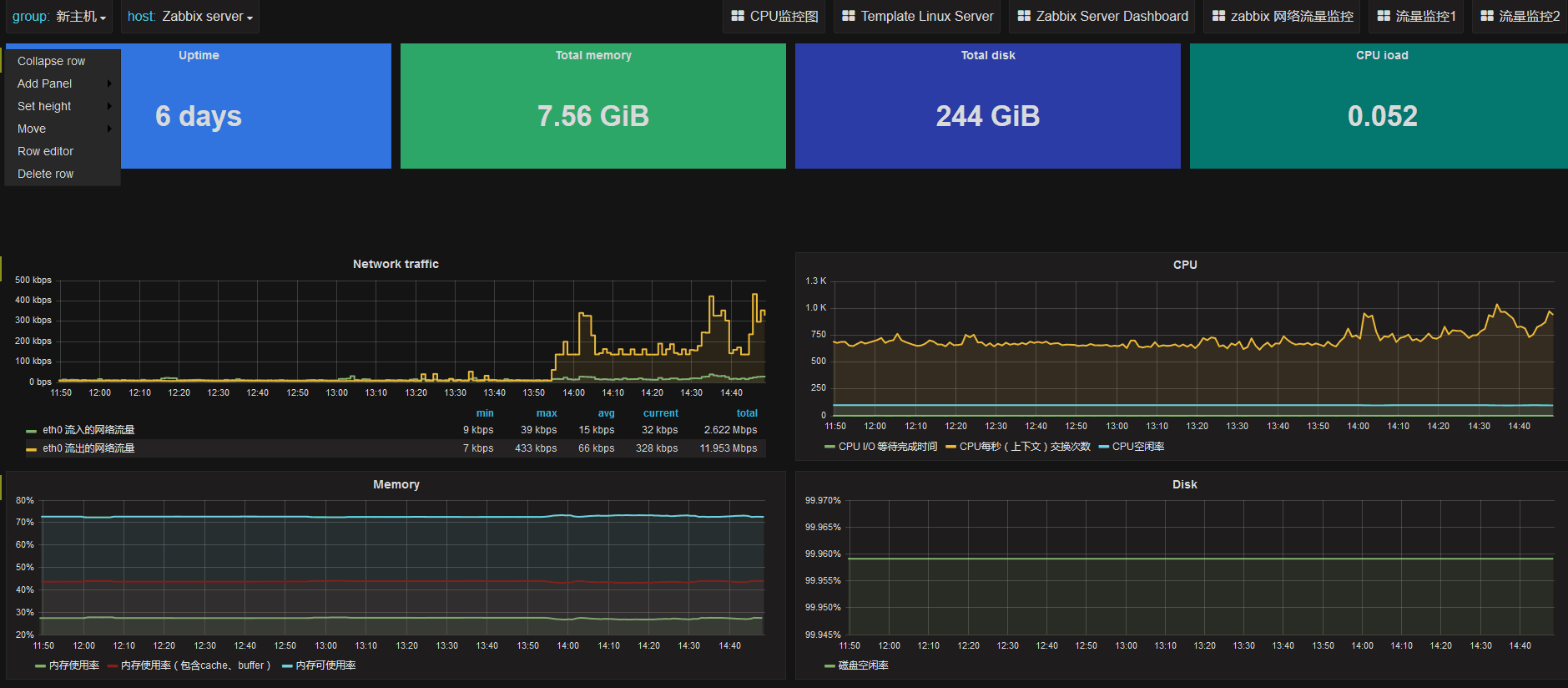
    下图是在仪表盘中使用模板功能在一个页面中用一个**graph**显示单台机器或所有机器的单个监控项或所有监控项。







    这个仪表盘模板是在一个页面中用多个graph显示单台机器的多个监控项



Rich Graphing

Fast and flexible client side graphs with a multitude of options.

* Click and select region to zoom
* Multiple Y-Axes
* Bars, Lines, Points
* Thresholds, Logarithmic scales
* Y-axis formats (bytes, milliseconds, etc)
* View or edit graph in fullscreen

Graph Styling

* Full control for how each series should be drawn
* Mix lines, bars and points
* Mix stacked series with isolated series
* Export any graph to png image (via UI or HTTP API)

grafana的配置文件：

    grafana后端的配置文件可以是多个以.ini结尾的配置文件，主要从三个配置文件读取配置：默认是$WORKING\_DIR/conf/defaults.ini，其次用户配置是$WORKING\_DIR/conf/custom.ini，用户配置则可以在命令行启动grafana时通过--config参数重新指定配置文件来覆盖。如果你是以deb或者rpm安装的，则默认的配置文件是/etc/grafana/grafana.ini，这个文件是在init.d的启动脚本中通过--config参数指定的。

    所有在配置文件中的配置都可以通过环境变量来覆盖，使用的语法如下：GF\_<SectionName>\_<KeyName>，例如：

**[html]** [view plain](http://blog.csdn.net/wzygis/article/details/52727067) [copy](http://blog.csdn.net/wzygis/article/details/52727067)

1. [security]
2. admin\_user = admin
4. [auth.google]
5. client\_secret = 0ldS3cretKey

如果使用环境变量，则是如下： 

**[html]** [view plain](http://blog.csdn.net/wzygis/article/details/52727067) [copy](http://blog.csdn.net/wzygis/article/details/52727067)

1. export GF\_SECURITY\_ADMIN\_USER=true
2. export GF\_AUTH\_GOOGLE\_CLIENT\_SECRET=newS3cretKey

下面具体看看每个配置段的配置：

**[html]** [view plain](http://blog.csdn.net/wzygis/article/details/52727067) [copy](http://blog.csdn.net/wzygis/article/details/52727067)

1. app\_mode：应用名称，默认是production
3. [path]
4. data：一个grafana用来存储sqlite3、临时文件、回话的地址路径
5. logs：grafana存储logs的路径
7. [server]
8. http\_addr：监听的ip地址，，默认是0.0.0.0
9. http\_port：监听的端口，默认是3000
10. protocol：http或者https，，默认是http
11. domain：这个设置是root\_url的一部分，当你通过浏览器访问grafana时的公开的domian名称，默认是localhost
12. enforce\_domain：如果主机的header不匹配domian，则跳转到一个正确的domain上，默认是false
13. root\_url：这是一个web上访问grafana的全路径url，默认是%(protocol)s://%(domain)s:%(http\_port)s/
14. router\_logging：是否记录web请求日志，默认是false
15. cert\_file：如果使用https则需要设置
16. cert\_key：如果使用https则需要设置
18. [database]
19. grafana默认需要使用数据库存储用户和dashboard信息，默认使用sqlite3来存储，你也可以换成其他数据库
20. type：可以是mysql、postgres、sqlite3，默认是sqlite3
21. path：只是sqlite3需要，定义sqlite3的存储路径
22. host：只是mysql、postgres需要，默认是127.0.0.1:3306
23. name：grafana的数据库名称，默认是grafana
24. user：连接数据库的用户
25. password：数据库用户的密码
26. ssl\_mode：只是postgres使用

29. [security]
30. admin\_user：grafana默认的admin用户，默认是admin
31. admin\_password：grafana admin的默认密码，默认是admin
32. login\_remember\_days：多少天内保持登录状态
33. secret\_key：保持登录状态的签名
34. disable\_gravatar：

37. [users]
38. allow\_sign\_up：是否允许普通用户登录，如果设置为false，则禁止用户登录，默认是true，则admin可以创建用户，并登录grafana
39. allow\_org\_create：如果设置为false，则禁止用户创建新组织，默认是true
40. auto\_assign\_org：当设置为true的时候，会自动的把新增用户增加到id为1的组织中，当设置为false的时候，新建用户的时候会新增一个组织
41. auto\_assign\_org\_role：新建用户附加的规则，默认是Viewer，还可以是Admin、Editor

44. [auth.anonymous]
45. enabled：设置为true，则开启允许匿名访问，默认是false
46. org\_name：为匿名用户设置组织名称
47. org\_role：为匿名用户设置的访问规则，默认是Viewer

50. [auth.github]
51. 针对github项目的，很明显，呵呵
52. enabled = false
53. allow\_sign\_up = false
54. client\_id = some\_id
55. client\_secret = some\_secret
56. scopes = user:email
57. auth\_url = https://github.com/login/oauth/authorize
58. token\_url = https://github.com/login/oauth/access\_token
59. api\_url = https://api.github.com/user
60. team\_ids =
61. allowed\_domains =
62. allowed\_organizations =

65. [auth.google]
66. 针对google app的，呵呵
67. enabled = false
68. allow\_sign\_up = false
69. client\_id = some\_client\_id
70. client\_secret = some\_client\_secret
71. scopes = https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email
72. auth\_url = https://accounts.google.com/o/oauth2/auth
73. token\_url = https://accounts.google.com/o/oauth2/token
74. api\_url = https://www.googleapis.com/oauth2/v1/userinfo
75. allowed\_domains =

78. [auth.basic]
79. enabled：当设置为true，则http api开启基本认证

82. [auth.ldap]
83. enabled：设置为true则开启LDAP认证，默认是false
84. config\_file：如果开启LDAP，指定LDAP的配置文件/etc/grafana/ldap.toml

87. [auth.proxy]
88. 允许你在一个HTTP反向代理上进行认证设置
89. enabled：默认是false
90. header\_name：默认是X-WEBAUTH-USER
91. header\_property：默认是个名称username
92. auto\_sign\_up：默认是true。开启自动注册，如果用户在grafana DB中不存在
94. [analytics]
95. reporting\_enabled：如果设置为true，则会发送匿名使用分析到stats.grafana.org，主要用于跟踪允许实例、版本、dashboard、错误统计。默认是true
96. google\_analytics\_ua\_id：使用GA进行分析，填写你的GA ID即可

99. [dashboards.json]
100. 如果你有一个系统自动产生json格式的dashboard，则可以开启这个特性试试
101. enabled：默认是false
102. path：一个全路径用来包含你的json dashboard，默认是/var/lib/grafana/dashboards

105. [session]
106. provider：默认是file，值还可以是memory、mysql、postgres
107. provider\_config：这个值的配置由provider的设置来确定，如果provider是file，则是data/xxxx路径类型，如果provider是mysql，则是user:password@tcp(127.0.0.1:3306)/database\_name，如果provider是postgres，则是user=a password=b host=localhost port=5432 dbname=c sslmode=disable
108. cookie\_name：grafana的cookie名称
109. cookie\_secure：如果设置为true，则grafana依赖https，默认是false
110. session\_life\_time：session过期时间，默认是86400秒，24小时

113. 以下是官方文档没有，配置文件中有的
114. [smtp]
115. enabled = false
116. host = localhost:25
117. user =
118. password =
119. cert\_file =
120. key\_file =
121. skip\_verify = false
122. from\_address = admin@grafana.localhost
124. [emails]
125. welcome\_email\_on\_sign\_up = false
126. templates\_pattern = emails/\*.html

129. [log]
130. mode：可以是console、file，默认是console、file，也可以设置多个，用逗号隔开
131. buffer\_len：channel的buffer长度，默认是10000
132. level：可以是"Trace", "Debug", "Info", "Warn", "Error", "Critical"，默认是info
134. [log.console]
135. level：设置级别
137. [log.file]
138. level：设置级别
139. log\_rotate：是否开启自动轮转
140. max\_lines：单个日志文件的最大行数，默认是1000000
141. max\_lines\_shift：单个日志文件的最大大小，默认是28，表示256MB
142. daily\_rotate：每天是否进行日志轮转，默认是true
143. max\_days：日志过期时间，默认是7,7天后删除

The port to bind to, defaults to 3000. To use port 80 you need to either give the Grafana binary permission for example:

$ sudo setcap 'cap\_net\_bind\_service=+ep' /usr/sbin/grafana-server

Or redirect port 80 to the Grafana port using:

$ sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3000

5. nginx 反向代理的配置。

由于grafana程序是静态文件，不需要PHP环境，程序默认端口是3000，可以修改成80端口。如果服务器的80端口被占用，可以用nginx 反向代理让grafana跑在80端口上面。

1）如果grafana的url不包含grafana目录，即输入120.zmzblog.com即可访问。

首先修改grafana的配置文件。

|  |  |
| --- | --- |
| 1  2 | [server]  domain = 120.zmzblog.com |

然后在nginx vhosts里面添加下面的虚拟主机即可。

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | server {          server\_name 120.zmzblog.com;          listen 80;          charset utf-8;            access\_log      /data/logs/nginx/120.zmzblog.com-access\_log main;          error\_log       /data/logs/nginx/120.zmzblog.com-error\_log  warn;        location /    {               proxy\_pass http://localhost:3000;      }    } |

Tên đầy đủ của [API](http://thietkeweb.center/) trong tiếng Anh là application programming interface. API còn được hiểu là một giao diện giữa phần mềm với phầm mềm. API là cách để các phần mềm (hệ điều hành, ứng dụng, các module trong hệ thống doanh nghiệp…) giao tiếp với nhau và tận dụng năng lực của nhau. Sở dĩ nói như vậy là bởi vì thế giới điện toán đòi hỏi rất nhiều phần mềm và bất cứ một phần mềm nào khi sử dụng, kể cả đó là một trang wed, ứng dụng di động hay game, thì trải nghiệm của bạn thực chất là sự kết hợp thành quả của ngành công nghiệp phần mềm.

