Grafana基础使用手册

Grafana基础使用手册

1.概述

Grafana是一个开源指标分析和可视化套件。它最常用于可视化基础设施性能数据和应用程序分析的时间序列数据。Grafana也应用于基它领域，包括工业传感器，家庭自动化，天气和过程控制。

当然作为IT工程师来讲，无非最关注的是如何把数据进行聚合后进行展示，例如：服务器性能数据、应用程序数据分析、elasticsearch、大数据分析数据等。那此时选择Grafana就是一个及好的选择，因为它支持多种数据来源。我们以前使用开源监控系统zabbix，但后来发现，如果要把监控数据和日志分析系统统一进行展示和分析时，就需要开发一套系统来对接。现在有了grafana过后变的相当Easy！

2.安装

grafana支持在多种环境下安装，包括windows、centos、ubuntu、mac、docker等环境。我将grafana安装在centos环境下。下面是安装方法。

支持以下安装方式

1.在线下载grafana

Description	Download
Stable for CentOS / Fedora / OpenSuse / Redhat Linux	4.1.2 (x86-64 rpm)

2.yum 远程安装稳定版本

```
yum install https://grafanarel.s3.amazonaws.com/builds/grafana-4.1.2-1486989747.x86_64.rpm
```

Install Stable

You can install Grafana using Yum directly.

```
$ sudo yum install https://grafanarel.s3.amazonaws.com/builds/grafana-4.1.2-1486989747.x86_64.rpm
```

3.使用yum仓库方式

保存以下信息到/etc/yum.repos.d/grafana.repo

```
[grafana]
name=grafana
baseurl=https://packagecloud.io/grafana/stable/el/6/$basearch
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packagecloud.io/gpg.key https://grafanarel.s3.amazonaws.com/RPM-GPG-KEY-grafana
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

使用yum install grafana -y

安装包的基本细节

二进制文件安装到 /usr/sbin/grafana-server
启动脚本复制到 /etc/init.d/grafana-server
环境变量文件 /etc/sysconfig/grafana-server
配置文件 /etc/grafana/grafana.ini
systemd服务名 grafana-server.service

默认日志文件 /var/log/grafana/grafana.log
默认数据库文件 /var/lib/grafana/grafana.db

启动服务

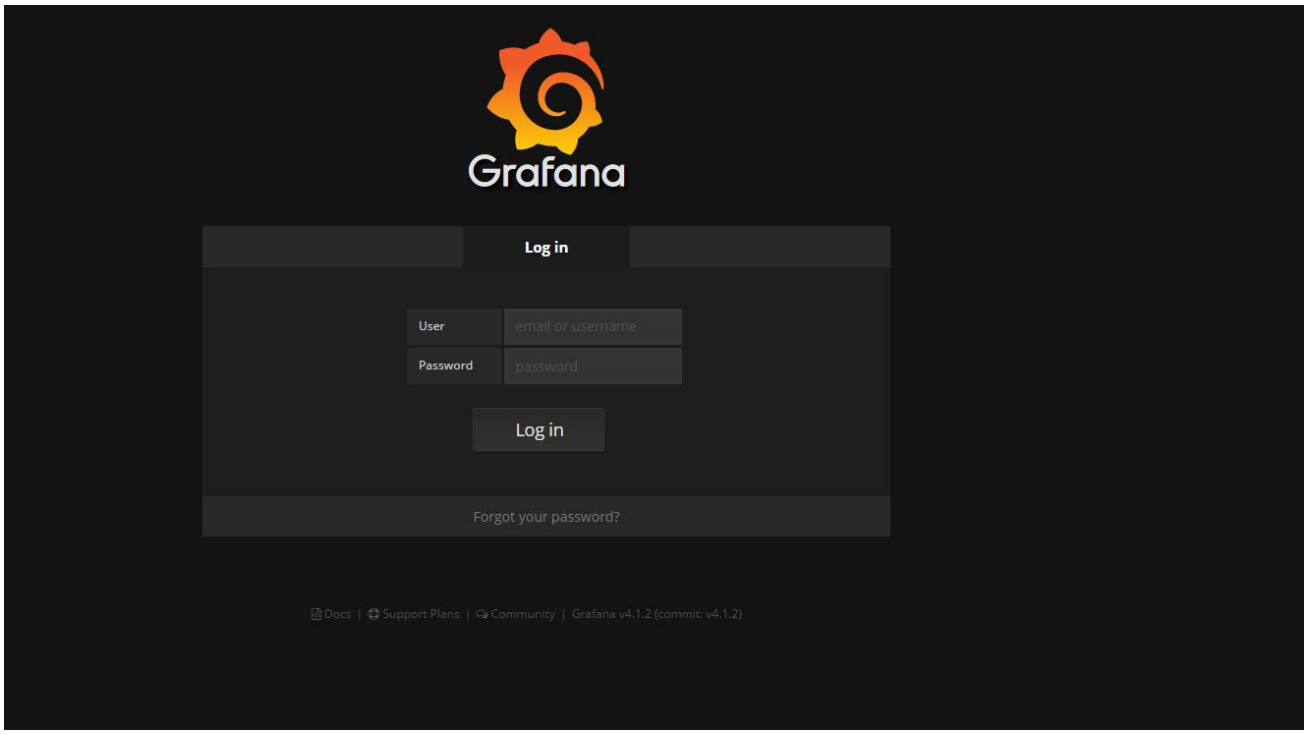
centos 6:
/etc/init.d/grafana-server start

centos 7:
systemctl enable grafana-server
systemctl start grafana-server

3.基础介绍

登录：http://x.x.x.x:3000 grafana默认启动端口为3000，可以在配置文件的[server]配置中修改grafana的相关配置信息

默认用户名和密码： admin admin



4.配置文件

grafana默认配置文件/etc/grafana/grafana.ini,那下面我们就来看一下grafana的基础配置。
以下配置文件中， ";" 注释的配置项表示默认值，如果需要修改请把 ";" 去除修改配置即可。

以下为主要配置文件：

#################### Grafana Configuration Example ####################
#

```
# Everything has defaults so you only need to uncomment things you want to
# change

# possible values : production, development
; app_mode = production

#实例名称，如要HOSTNAME 变量为空，默认就为主机名
; instance_name = ${HOSTNAME}


#################################### Paths ############################
[paths]
# 默认数据存储目录如，db，session等
#
;data = /var/lib/grafana
#
#日志存储目录
#
;logs = /var/log/grafana
#
#插件安装目录，Grafana会自动扫描并查找插件
#
;plugins = /var/lib/grafana/plugins


[server]
# 协议 (http or https)
;protocol = http

# 绑定的IP地址，为空将在所有接口上绑定
;http_addr =

# 默认端口
;http_port = 3000

# 对外的域名
;domain = localhost

# 如果主机头与域不匹配，则重定向到正确的域
#防止DNS重新绑定攻击
;enforce_domain = false

# 在浏览器中使用的完全公开的网址，用于重定向和电子邮件
# 如果使用反向代理和子路径指定完整url（带子路径）
;root_url = http://localhost:3000

# web 请求日志
;router_logging = false

#相对路径
;static_root_path = public
```

```
# gzip压缩
;enable_gzip = false

# https证书
;cert_file =
;cert_key =

#################################### Database #############
[database]
# 默认数据库配置
# as seperate properties or as on string using the url propertie.

# Either "mysql", "postgres" or "sqlite3", it's your choice
;type = sqlite3
;host = 127.0.0.1:3306
;name = grafana
;user = root
# If the password contains # or ; you have to wrap it with trippel quotes. Ex """#password;"""
;password =

# 使用URL或以前的字段来配置数据库
# Example: mysql://user:secret@host:port/database
;url =

# For "postgres" only, either "disable", "require" or "verify-full"
;ssl_mode = disable

# 仅适用于sqlite3数据库。将存储数据库的文件路径
;path = grafana.db

#################################### Security ############
[security]
# 系统默认用户为admin，在启动时创建
;admin_user = admin

# 默认admin密码，可以在第一次启动grafana之前更改，或在配置文件中设置
;admin_password = admin

# used for signing
;secret_key = SW2YcwTIb9zpOOhoPsMm

# 自动登录记住天数
;login_remember_days = 7
;cookie_username = grafana_user
;cookie_remember_name = grafana_remember

# 禁用用户头像
;disable_gravatar = false

# 数据源代理白名单（ip_or_domain：端口以空格分隔）
;data_source_proxy_whitelist =
```

```
#################################### Users ########
[users]
# 禁用用户注册
allow_sign_up = false

# 允许非管理员创建组织
;allow_org_create = true

# 如果为true，将自动将新用户分配到 (id为1)的组织
;auto_assign_org = true

# 新用户默认角色
;auto_assign_org_role = Viewer

# 登录页面的html字段说明
;login_hint = email or username

# 默认UI主题 ("dark" or "light")
;default_theme = dark

[auth]
# 可禁用登录表单，如果使用OAuth登录时非常有用
;disable_login_form = false

#################################### SMTP / Emailing ############
[smtp]
;enabled = false
;host = localhost:25
;user =
;password =
;cert_file =
;key_file =
;skip_verify = false
;from_address = admin@grafana.localhost

[emails]
;welcome_email_on_sign_up = false

#################################### Alerting ##################
[alerting]
# Makes it possible to turn off alert rule execution.
;execute_alerts = true
```

5.数据源

Grafana支持以时间序列存储数据的相关数据源，每个数据源都有一个特定的查询编辑器，该编辑器针对特定数据源公开的功能进行自定义。当前grafana支持以下数据源：Graphite、InfluxDB、OpenTSDB、Promethenus、Elasticsearch、CloudWatch。每个数据源的查询语言和功能是不同的，Grafana可以将不同尖的数据合并到单个Dashboard中。但每个面板都要与属于特定组织的数据源相关联。换句话说就是每个panel只能使用属于当前组织的数据源。

下面以使用influxdb为例：

1.安装influxdb

配置yum

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF
```

安装

```
yum install influxdb
```

配置

默认配置文件为：/etc/influxdb/influxdb.conf

```
[admin]
  # Determines whether the admin service is enabled.
  enabled = true        #开启admin管理

  # The default bind address used by the admin service.
  bind-address = ":8083"

  # Whether the admin service should use HTTPS.
  # https-enabled = false

  # The SSL certificate used when HTTPS is enabled.
  # https-certificate = "/etc/ssl/influxdb.pem"

[http]
  # Determines whether HTTP endpoint is enabled.
  enabled = true

  # The bind address used by the HTTP service.
  bind-address = ":8086"

  # Determines whether HTTP authentication is enabled.
  auth-enabled = true  #开启用户验证 生产环境一定要注意

  # The default realm sent back when issuing a basic auth challenge.
  # realm = "InfluxDB"

  # Determines whether HTTP request logging is enable.d
  log-enabled = true

  # Determines whether detailed write logging is enabled.
  write-tracing = false

  # Determines whether the pprof endpoint is enabled.  This endpoint is used for
  # troubleshooting and monitoring.
  pprof-enabled = true
```

```
  # Determines whether HTTPS is enabled.
  https-enabled = false

  # The SSL certificate to use when HTTPS is enabled.
  https-certificate = "/etc/ssl/influxdb.pem"
```

启动

centos7

systemctl start influxdb

centos6

service influxdb start

创建数据库

```
[root@localhost ~]# influx -host '127.0.0.1' -port '8086'
Connected to http://127.0.0.1:8086 version 1.2.0
InfluxDB shell version: 1.2.0
>
>
> show databases;
name: databases
name
----
_internal

> create database grafana
> use grafana
Using database grafana
> create user "grafana" with  password 'grafana' with all privileges
>
```

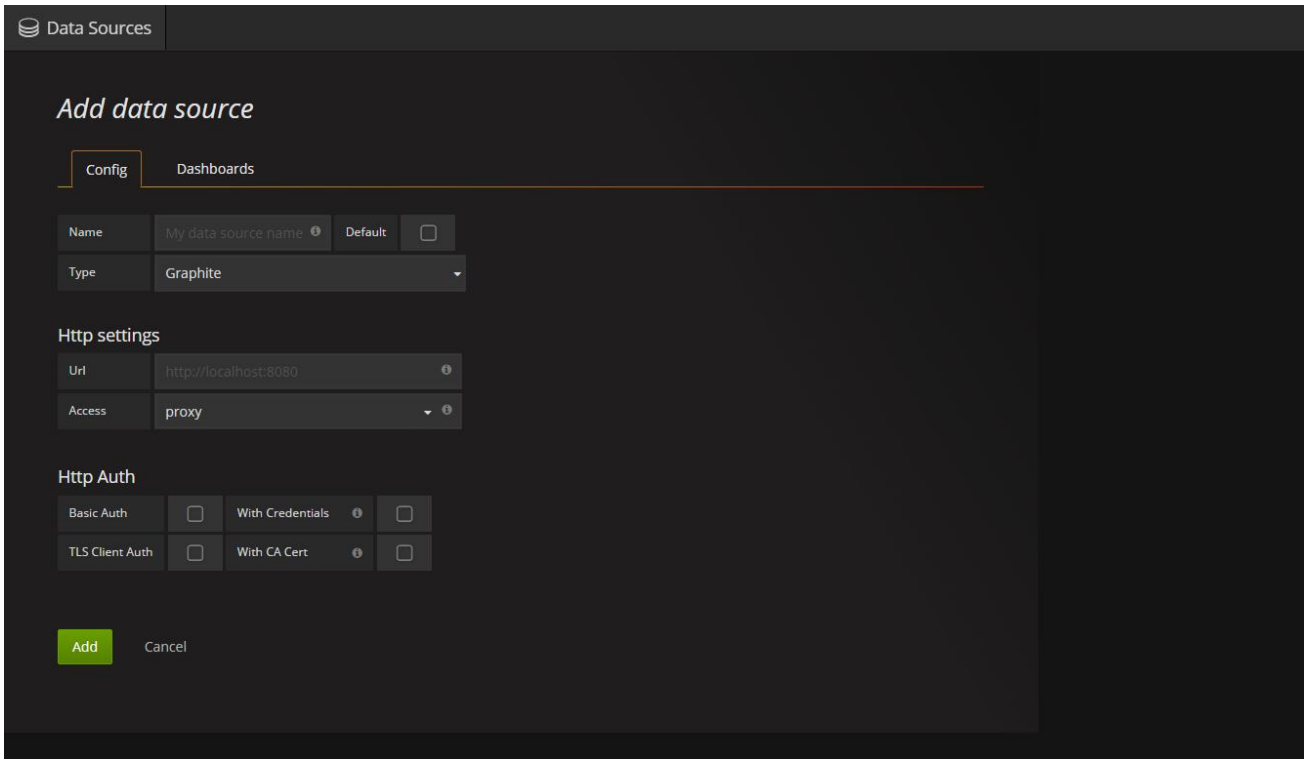
登录http://x.x.x.x:8083 输入用户名、密码、数据库、端口进行登录

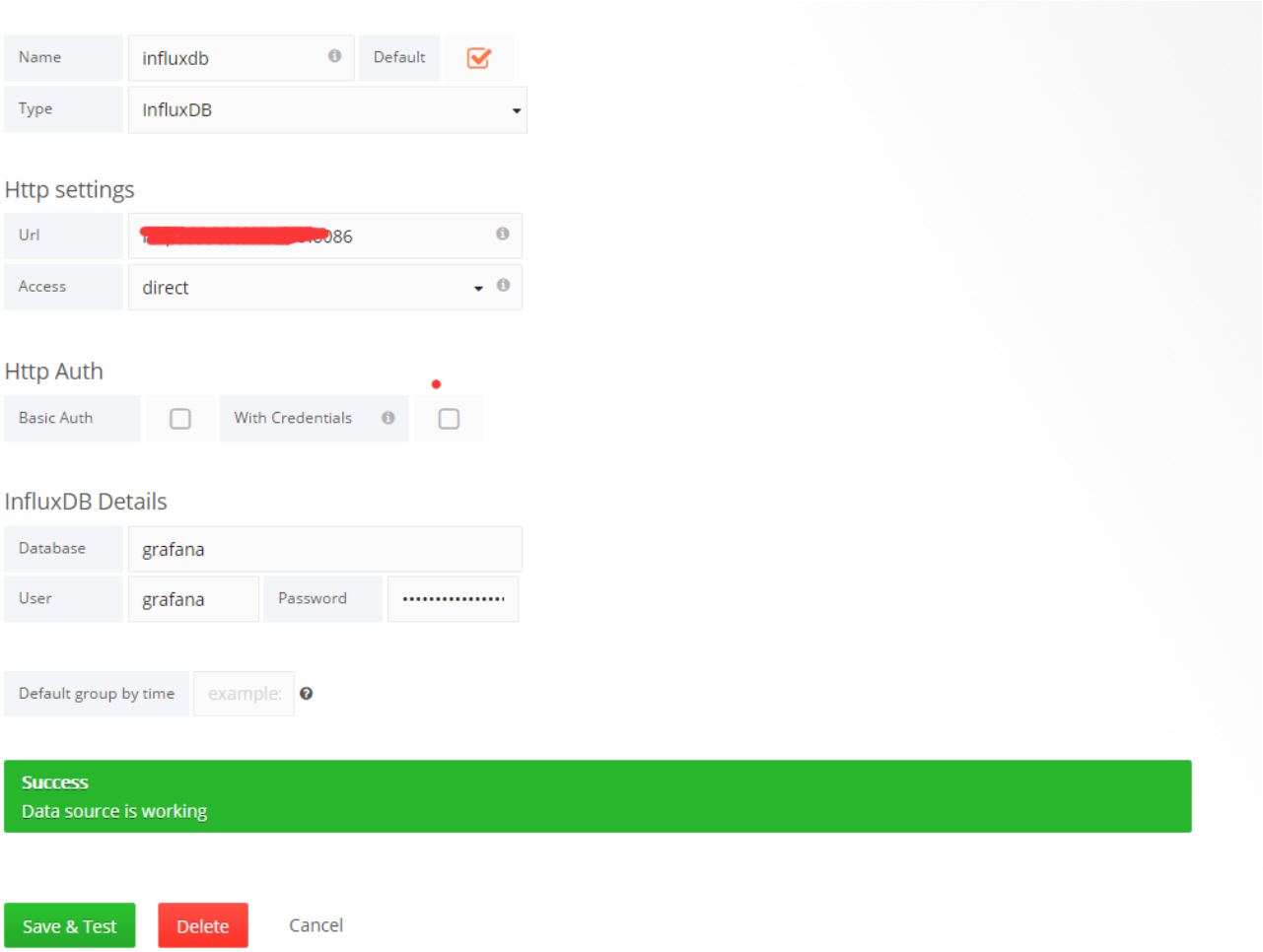
基本语法

SHOW DATABASES	查看所有的数据库
SHOW MEASUREMENTS	查看当前库中的表
SHOW TAG KEYS FROM "diskio"	查看表中的keys
SHOW USERS	查看Users
SHOW STATS	查看数据库状态
select * from httpd order by time desc limit 10	执行基础SQL语句

参考文档：https://docs.influxdata.com/influxdb/v1.2/introduction/installation/

添加数据源





6.Telegraf

Telegraf 是一个用 Go 编写的代理程序，可收集系统和服务的统计数据，并写入到 Influxdb 数据库。Telegraf 具有内存占用小的特点，通过插件系统开发人员可轻松添加支持其他服务的扩展。
目前已有的插件包括：(官方文档：https://docs.influxdata.com/telegraf/v1.2/)

- System (memory, CPU, network, etc.)
- Docker
- MySQL
- PostgreSQL
- Redis

支持系统：

- Ubuntu
- Redhat
- SLES
- FreeBSD
- MAC
- Windows

安装：

```
cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl = https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
```

gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
EOF

centos 6 :

yum install telegraf
service telegraf start

centos 7:

yum install telegraf
systemctl start telegraf

配置文件 :

```
[global_tags]
  # dc = "us-east-1" # will tag all metrics with dc=us-east-1
  # rack = "1a"
  ## Environment variables can be used as tags, and throughout the config file
  # user = "$USER"
[agent]
  ## Default data collection interval for all inputs
  interval = "10s"
  ## Rounds collection interval to 'interval'
  ## ie, if interval="10s" then always collect on :00, :10, :20, etc.
  round_interval = true
  ## Telegraf will send metrics to outputs in batches of at most
  ## metric_batch_size metrics.
  ## This controls the size of writes that Telegraf sends to output plugins.
  metric_batch_size = 1000
  ## For failed writes, telegraf will cache metric_buffer_limit metrics for each
  ## output, and will flush this buffer on a successful write. Oldest metrics
  ## are dropped first when this buffer fills.
  ## This buffer only fills when writes fail to output plugin(s).
  metric_buffer_limit = 10000
  ## Collection jitter is used to jitter the collection by a random amount.
  ## Each plugin will sleep for a random time within jitter before collecting.
  ## This can be used to avoid many plugins querying things like sysfs at the
  ## same time, which can have a measurable effect on the system.
  collection_jitter = "0s"
  ## Default flushing interval for all outputs. You shouldn't set this below
  ## interval. Maximum flush_interval will be flush_interval + flush_jitter
  flush_interval = "10s"
  ## Jitter the flush interval by a random amount. This is primarily to avoid
  ## large write spikes for users running a large number of telegraf instances.
  ## ie, a jitter of 5s and interval 10s means flushes will happen every 10-15s
  flush_jitter = "0s"
  ## By default, precision will be set to the same timestamp order as the
  ## collection interval, with the maximum being 1s.
  ## Precision will NOT be used for service inputs, such as logparser and statsd.
  ## Valid values are "ns", "us" (or "µs"), "ms", "s".
  precision = ""
  ## Logging configuration:
  ## Run telegraf with debug log messages.
```

```
  debug = false
  ## Run telegraf in quiet mode (error log messages only).
  quiet = false
  ## Specify the log file name. The empty string means to log to stderr.
  logfile = ""
  ## Override default hostname, if empty use os.Hostname()
  hostname = "ecs-ci-139.129.213.65"     #agent主机名可以采用salt来部署
  ## If set to true, do no set the "host" tag in the telegraf agent.
  omit_hostname = false
[[outputs.influxdb]]
  ## The full HTTP or UDP endpoint URL for your InfluxDB instance.
  ## Multiple urls can be specified as part of the same cluster,
  ## this means that only ONE of the urls will be written to each interval.
  # urls = ["udp://localhost:8089"] # UDP endpoint example
  urls = ["http://x.x.x.x:8086"]     # influxdb地址必填
  ## The target database for metrics (telegraf will create it if not exists).
  database = "grafana" # required            #influxdb数据库
  ## Retention policy to write to. Empty string writes to the default rp.
  retention_policy = ""
  ## Write consistency (clusters only), can be: "any", "one", "quorum", "all"
  write_consistency = "any"
  ## Write timeout (for the InfluxDB client), formatted as a string.
  ## If not provided, will default to 5s. 0s means no timeout (not recommended).
  timeout = "5s"
  username = "grafana"
  password = "xxxxxxxxx"
  ## Set the user agent for HTTP POSTs (can be useful for log differentiation)
  # user_agent = "telegraf"
  ## Set UDP payload size, defaults to InfluxDB UDP Client default (512 bytes)
  # udp_payload = 512
  ## Optional SSL Config
  # ssl_ca = "/etc/telegraf/ca.pem"
  # ssl_cert = "/etc/telegraf/cert.pem"
  # ssl_key = "/etc/telegraf/key.pem"
  ## Use SSL but skip chain & host verification
  # insecure_skip_verify = false
#[]以下为插件配置，如需要配置请取消#号即可
[[inputs.cpu]]
  ## Whether to report per-cpu stats or not
  percpu = true
  ## Whether to report total system cpu stats or not
  totalcpu = true
  ## If true, collect raw CPU time metrics.
  collect_cpu_time = false
[[inputs.disk]]
  ## By default, telegraf gather stats for all mountpoints.
  ## Setting mountpoints will restrict the stats to the specified mountpoints.
  # mount_points = ["/"]
  ## Ignore some mountpoints by filesystem type. For example (dev)tmpfs (usually
  ## present on /run, /var/run, /dev/shm or /dev).
  ignore_fs = ["tmpfs", "devtmpfs"]
[[inputs.diskio]]
```

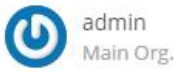
```
  ## By default, telegraf will gather stats for all devices including
  ## disk partitions.
  ## Setting devices will restrict the stats to the specified devices.
  # devices = ["sda", "sdb"]
  ## Uncomment the following line if you need disk serial numbers.
  # skip_serial_number = false
[[inputs.kernel]]
  # no configuration
[[inputs.mem]]
  # no configuration
[[inputs.processes]]
  # no configuration
[[inputs.swap]]
  # no configuration
[[inputs.system]]
  # no configuration
[[inputs.net]]
[[inputs.netstat]]
```

7.面板绘图

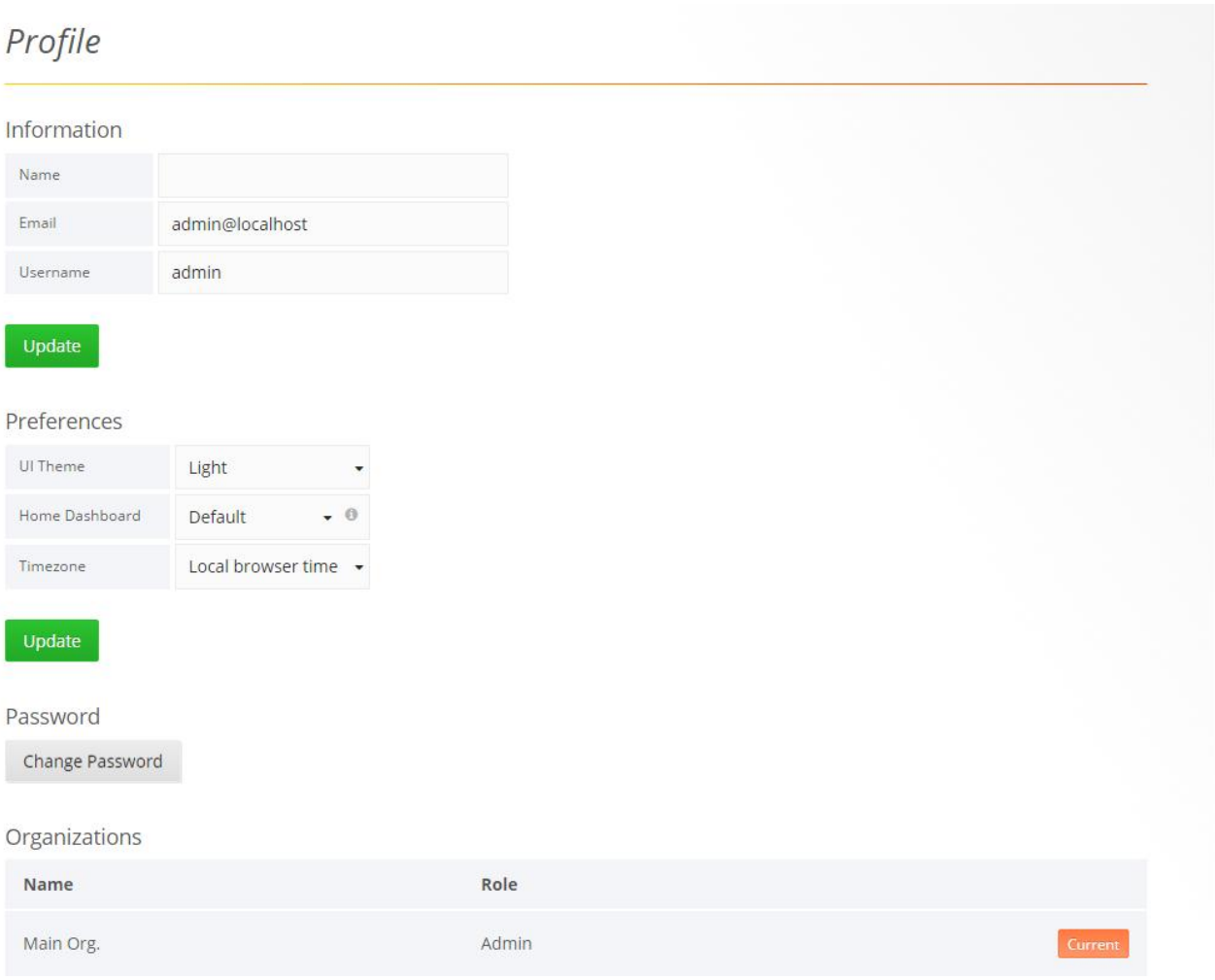
User and Org

用户：

默认登录系统是以admin用户进行登录，在帐号下是你所在的组织。

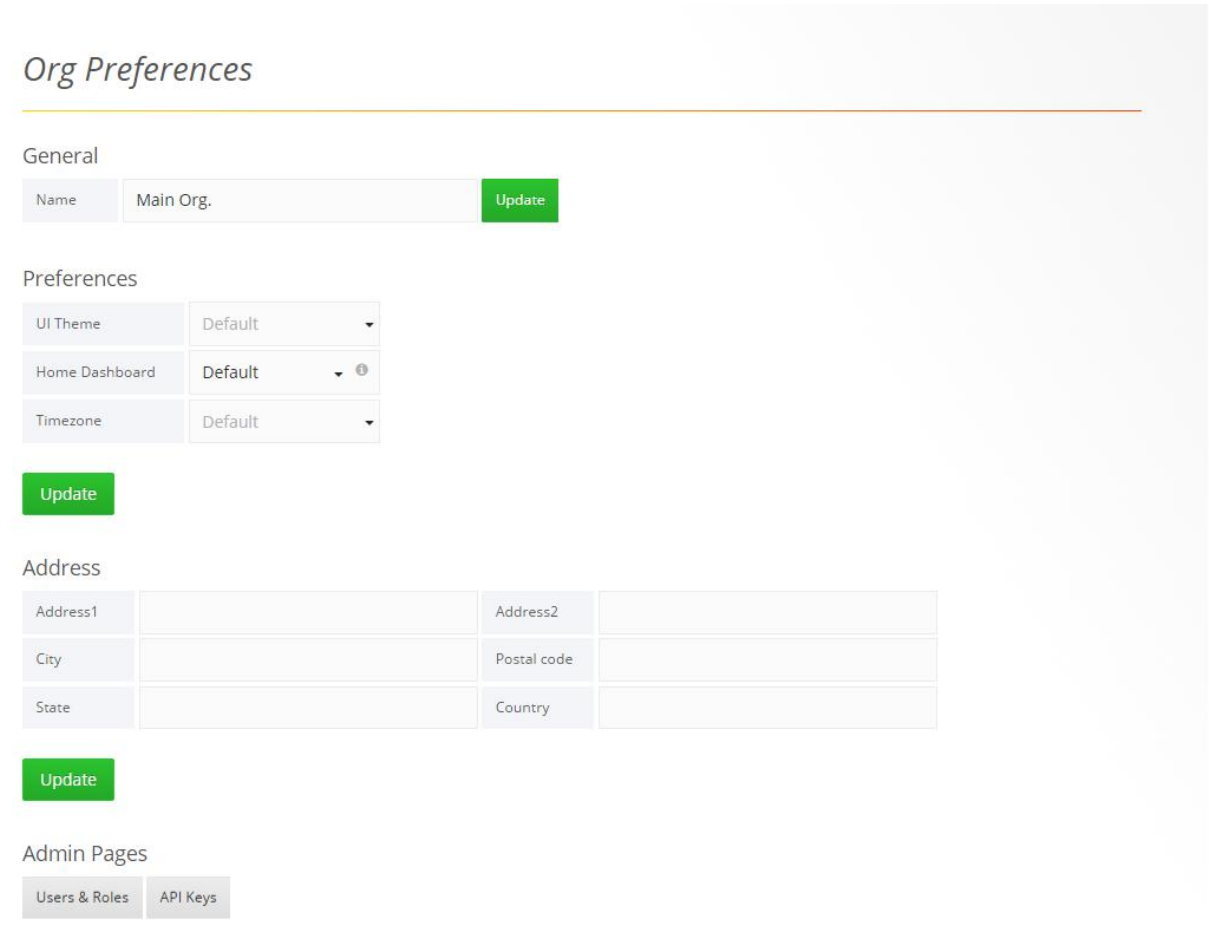


选择可以编辑当前用户的基本信息，同时也可以修改用户密码

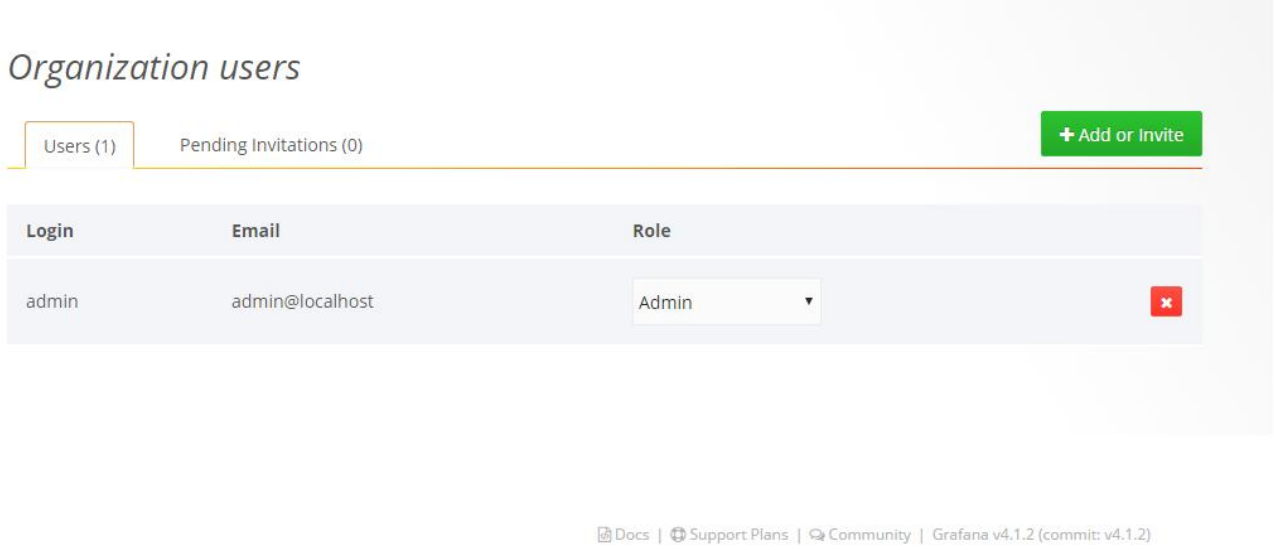


组织：

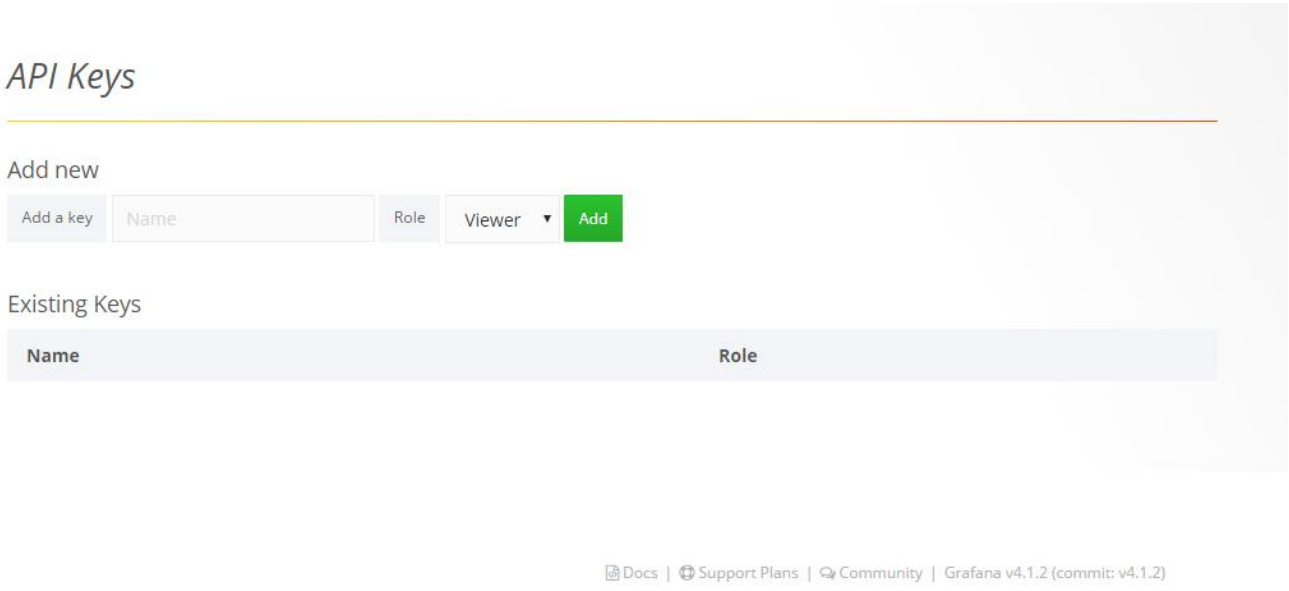
可以修改组织名，地址等基本信息



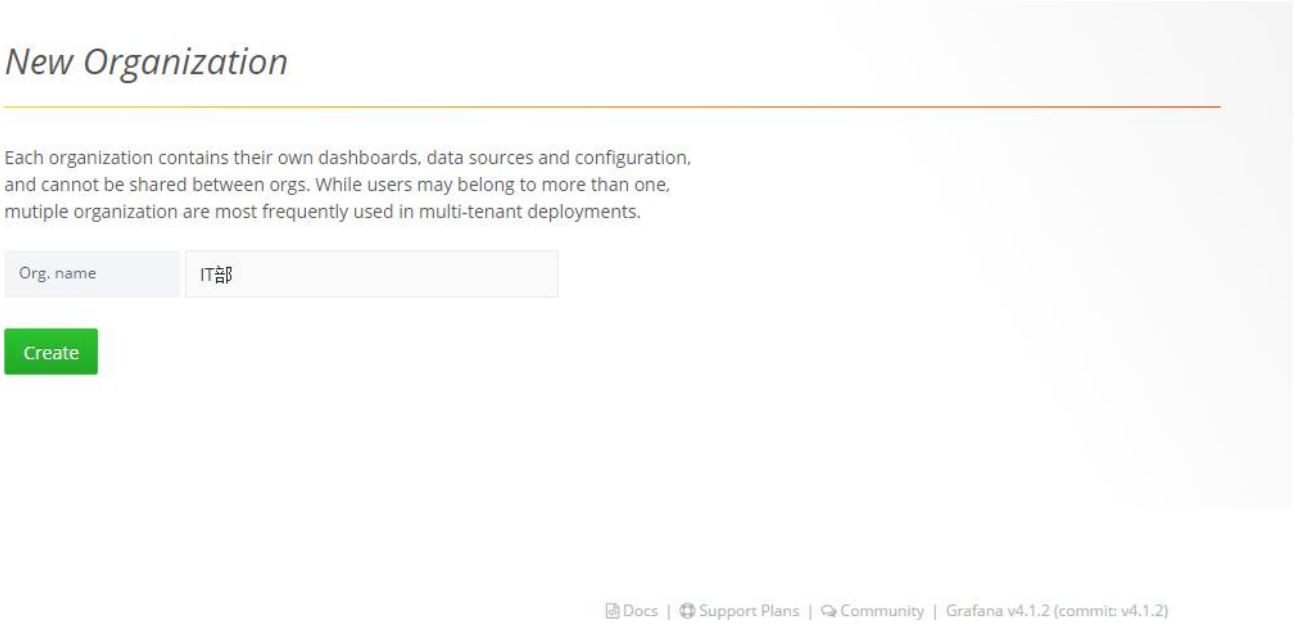
管理组织当中的用户，可配置相关角色权限。默认权限有：Admin、Viewer、Editor、Read Only Editor。Grafana有两个级别的管理，分别是组织管理员和Grafana管理员。后面后有相关介绍



在此可以设置grafana的相关API key

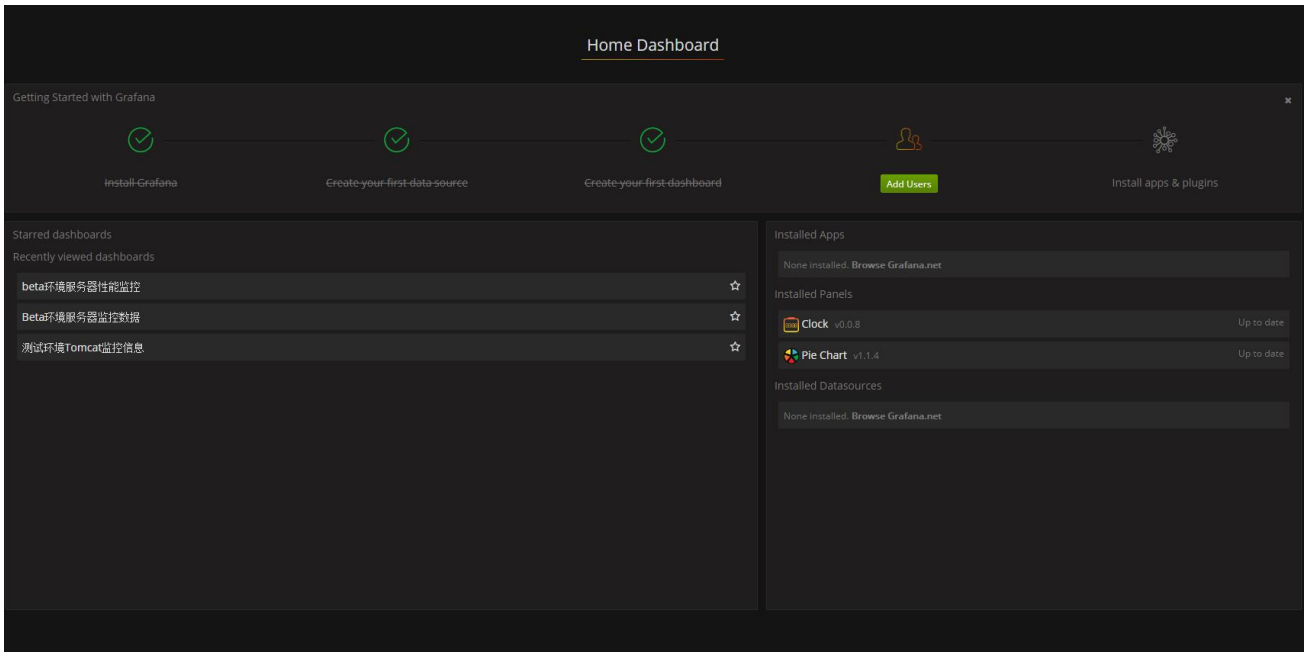


创建一个组织



Home DashBoard

是grafana的全局视图，可以认为是一组一个或多个面板的组成。可以通过右上角的仪表时间选择器进行控制。



上面是安装完grafana后需要做的操作流程

1.安装grafana

2.创建您的数据来源

3.创建一个dashborad

4.添加用户

5.安装apps或插件

Dashboard还包含如下信息

1.共享的Dashboard列表

2.最近查看的Dashboards列表

3.已安装的Apps

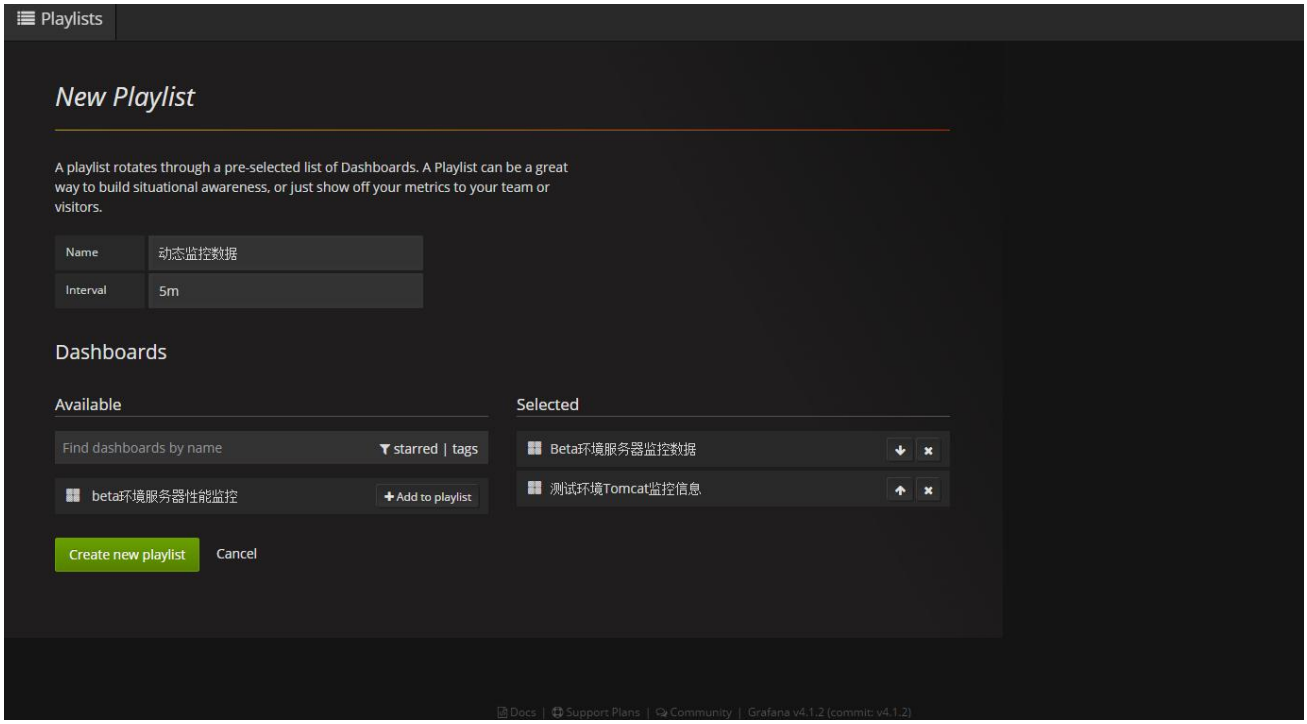
4.已安装的面板

如上图可以看出我们安装了两个面板：时间面板和饼图面板

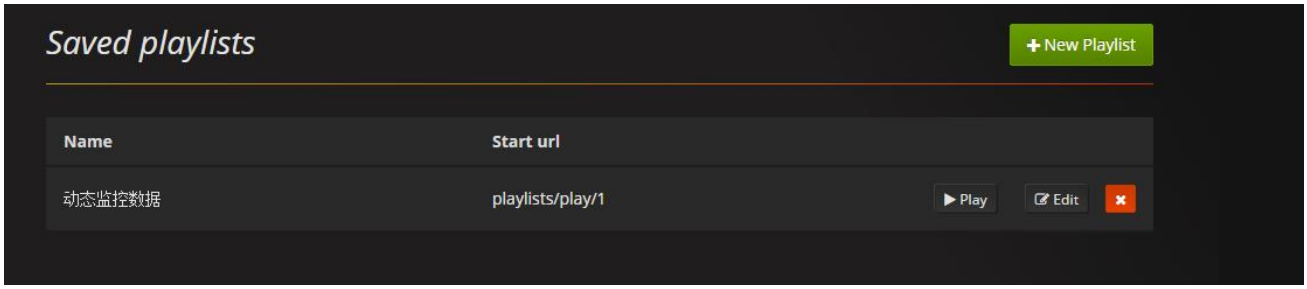
5.已安装的数据源

playlist（播放列表）

它是一种特殊类型的信息中心，可通过信息显示板列表进行滚动



点击play即可在两个Dashboards中进行依次展示，默认间隔时间为5分钟

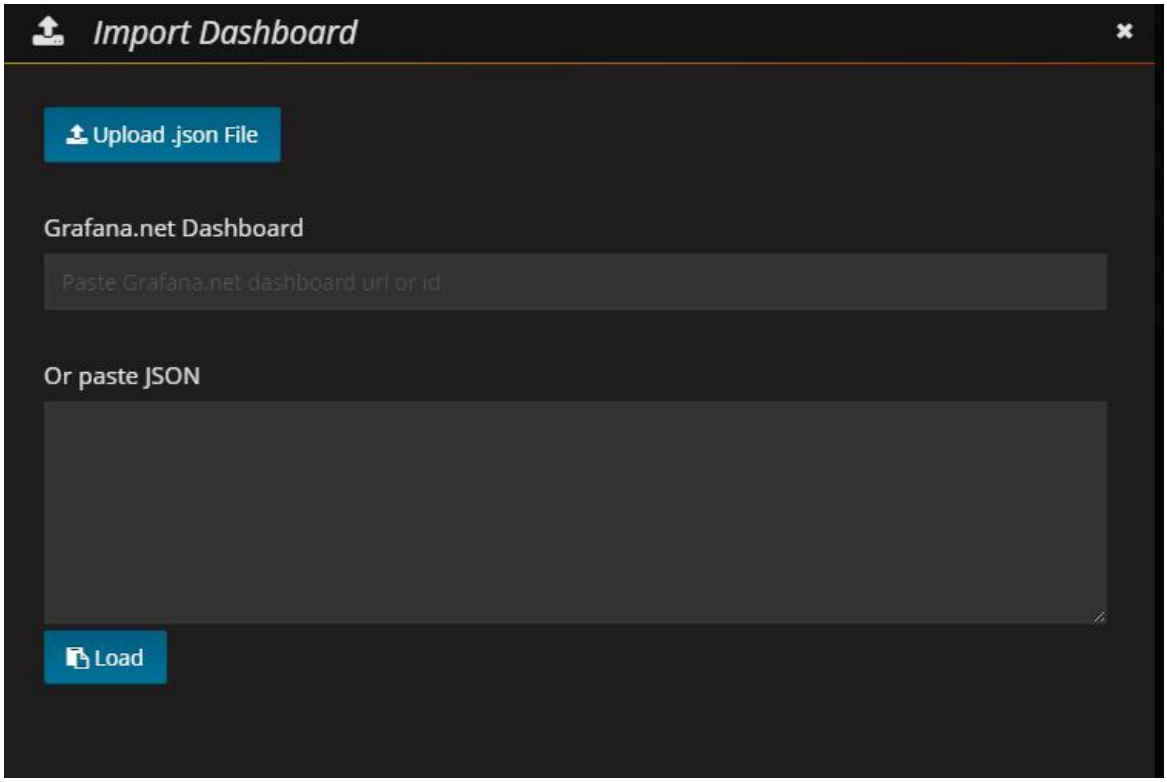


可以对播放列表进行控制

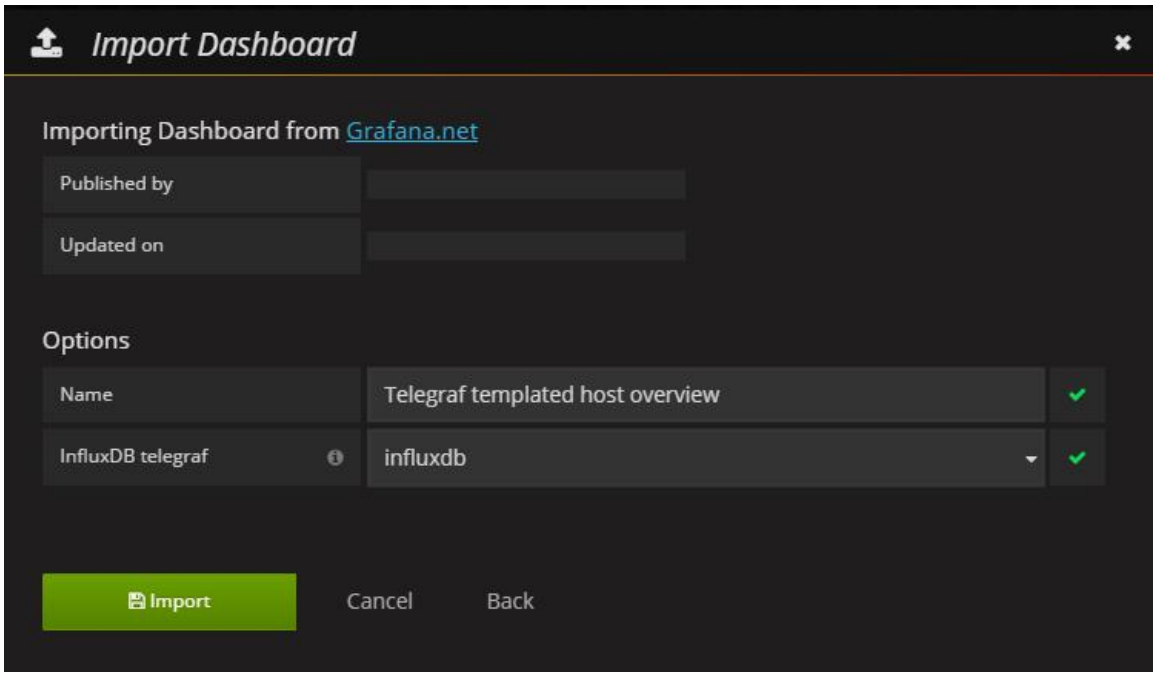
导入模板

下载模板
https://grafana.net/dashboards/928

导入
在Dashboards中选择Import-->选择json的模板文件

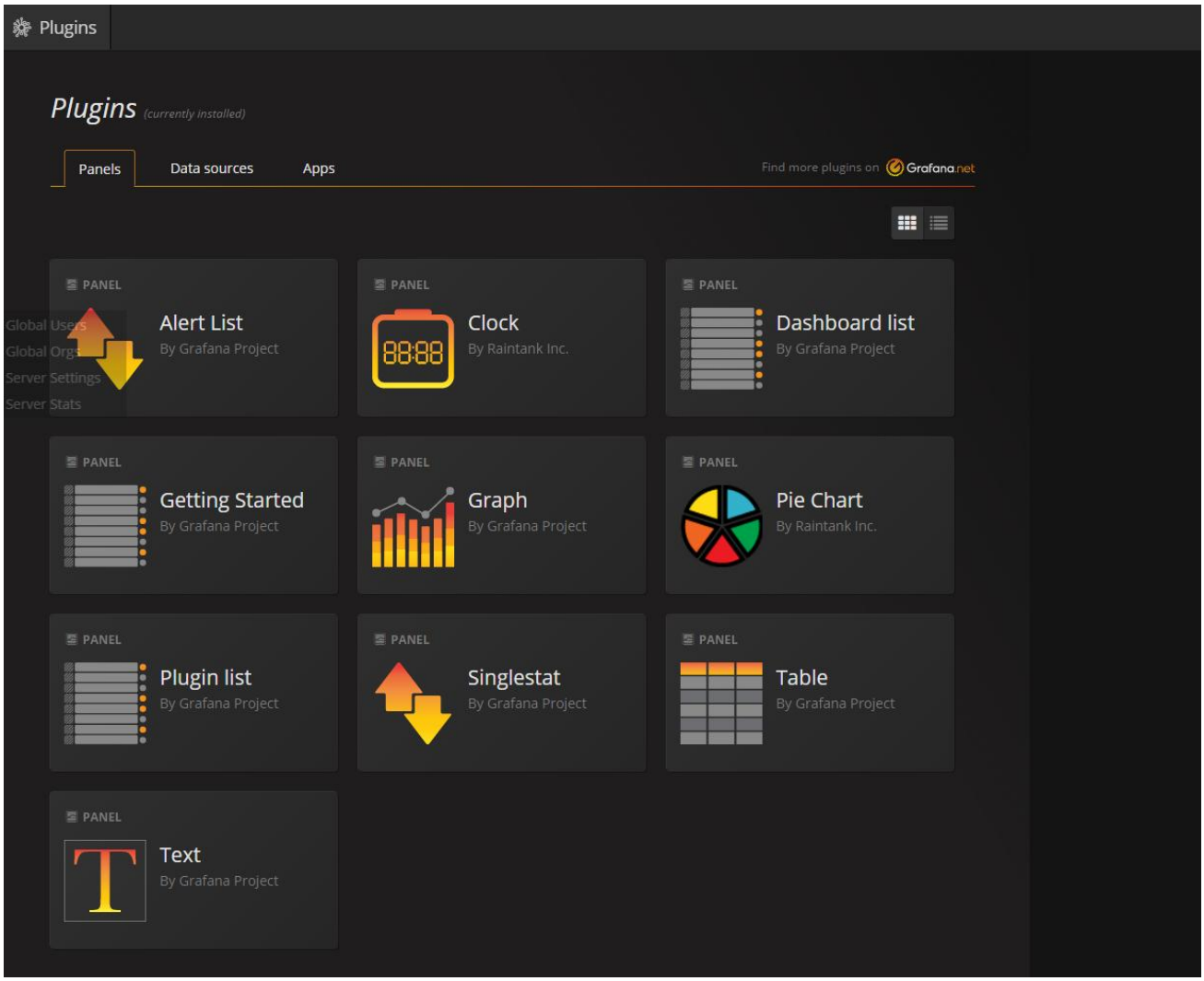

Import Dashboard
Upload .json File
Grafana.net Dashboard
Paste Grafana.net dashboard url or id
Or paste JSON
Load

导入模板

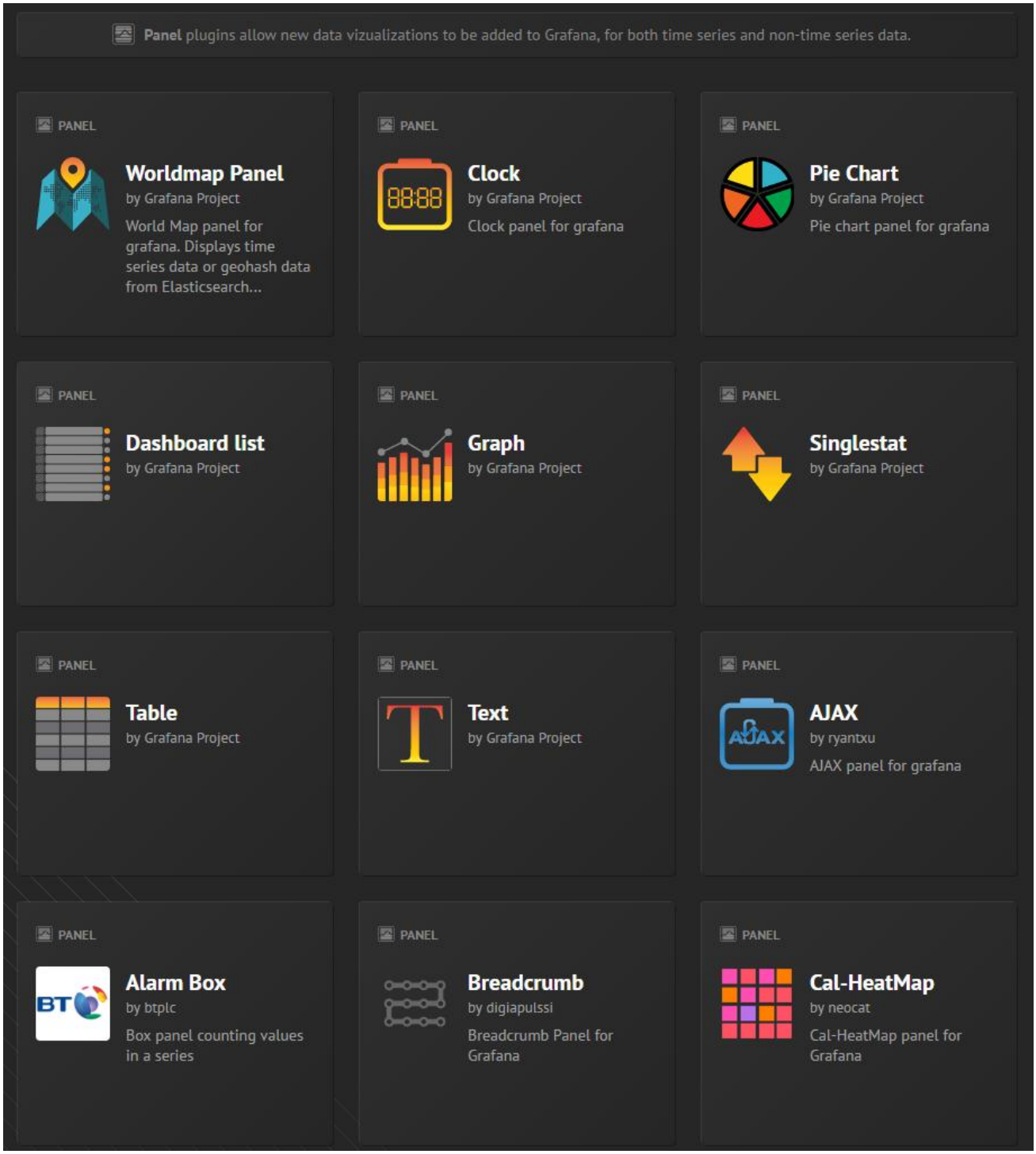

Import Dashboard
Importing Dashboard from Grafana.net
Published by
Updated on
Options
Name
Telegraf templated host overview
InfluxDB telegraf
influxdb
Import
Cancel
Back

结果

插件panels和apps



下载插件需要去官网下载您需要的具体panels和apps，官方支持plugins很多支持以下panel



如果需要下载就选中后按配置方法安装即可。以下以安装 worldmap panel为例。点击**install plugin**

All Plugins » Worldmap Panel

Worldmap Panel by Grafana Project

PANEL

World Map panel for grafana. Displays time series data or geohash data from Elasticsearch overlaid on a world map.

Overview | Installation

Install Plugin

Version

0.0.16

Worldmap Panel Plugin for Grafana

The Worldmap Panel is a tile map of the world that can be overlaid with circles representing data points from a query. It can be used with time series metrics, with geohash data from Elasticsearch or Table Data.

Dependencies:

GRAFANA 3.X.X

Links:

Project site
MIT License

执行安装步骤

Overview | Installation

Installing on a local Grafana:

For local instances, plugins are installed and updated via a simple CLI command.
Plugins are not updated automatically, however you will be notified when updates are available right within your Grafana.

1 Install the Panel

Use the grafana-cli tool to install Worldmap Panel from the commandline:

```
grafana-cli plugins install grafana-worldmap-panel
```

The plugin will be installed into your grafana plugins directory; the default is /var/lib/grafana/plugins. More information on the cli tool.

Note: Grafana 3.0 or greater is required to install and use plugins. Download Grafana latest.

2 Add the Panel to a Dashboard

Installed panels are available immediately in the Dashboards section in your Grafana main menu, and can be added like any other core panel in Grafana.

To see a list of installed panels, click the **Plugins** item in the main menu. Both core panels and installed panels will appear.

下载插件目录位于/var/lib/grafana/plugins

[root@localhost ~]# grafana-cli plugins install grafana-worldmap-panel

installing grafana-worldmap-panel @ 0.0.16

from url: https://grafana.net/api/plugins/grafana-worldmap-panel/versions/0.0.16/download

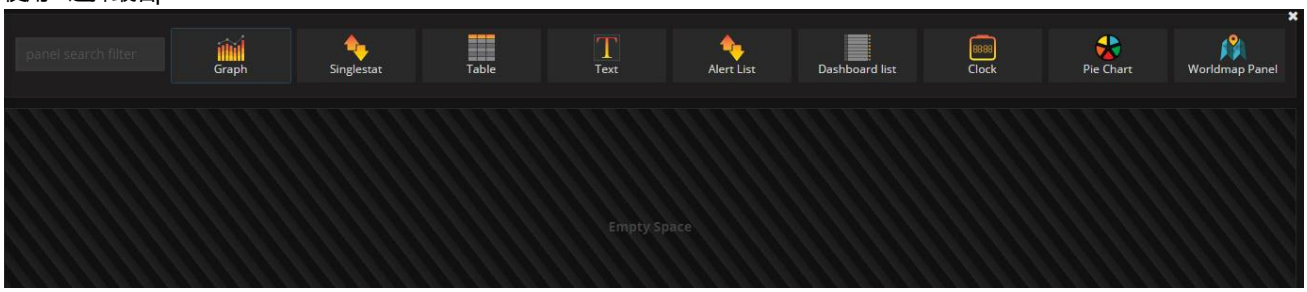
into: /var/lib/grafana/plugins

✔ Installed grafana-worldmap-panel successfully

Restart grafana after installing plugins . <service grafana-server restart>

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

使用：选择最后panel

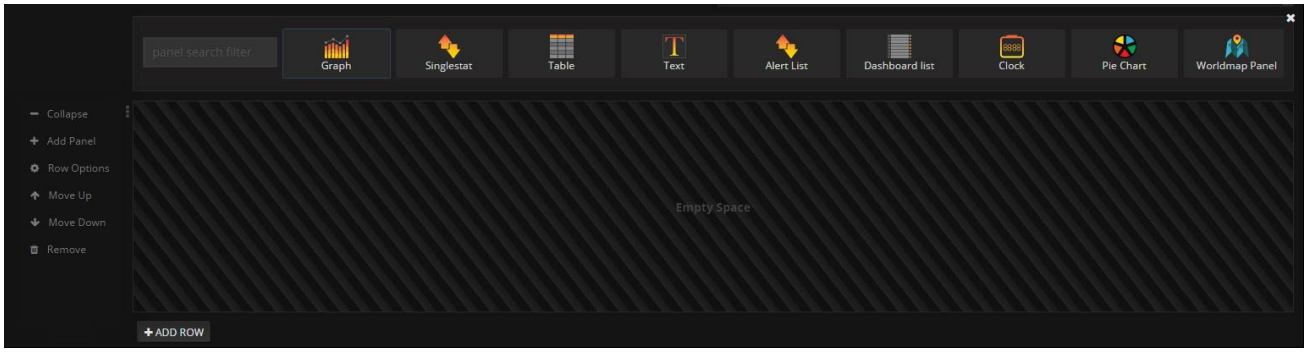


8.自定义绘图

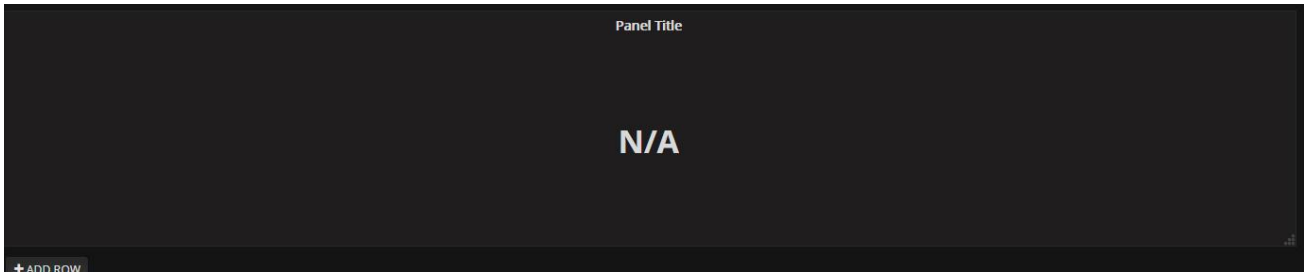
下面我们在一个新的DashBoard中创建以下面板信息,那建立以下信息需要安装一个clock plugin用来建立时间面板。

在创建这边面板之前，我们来看到一个Dashboard是如何组成的。Dashboard由多个Row组成，在Row中创建面板，一行Row分为12列，面板默认Span为12。可以自定每个面板的Span宽度。也可以自定义高度等。

添加Row时需要添加指定类型的面板，那Uptime 我们数据是要从数据源中获取。就需要选择Singlestat，如果需要绘线性图表就需要选择Graph。

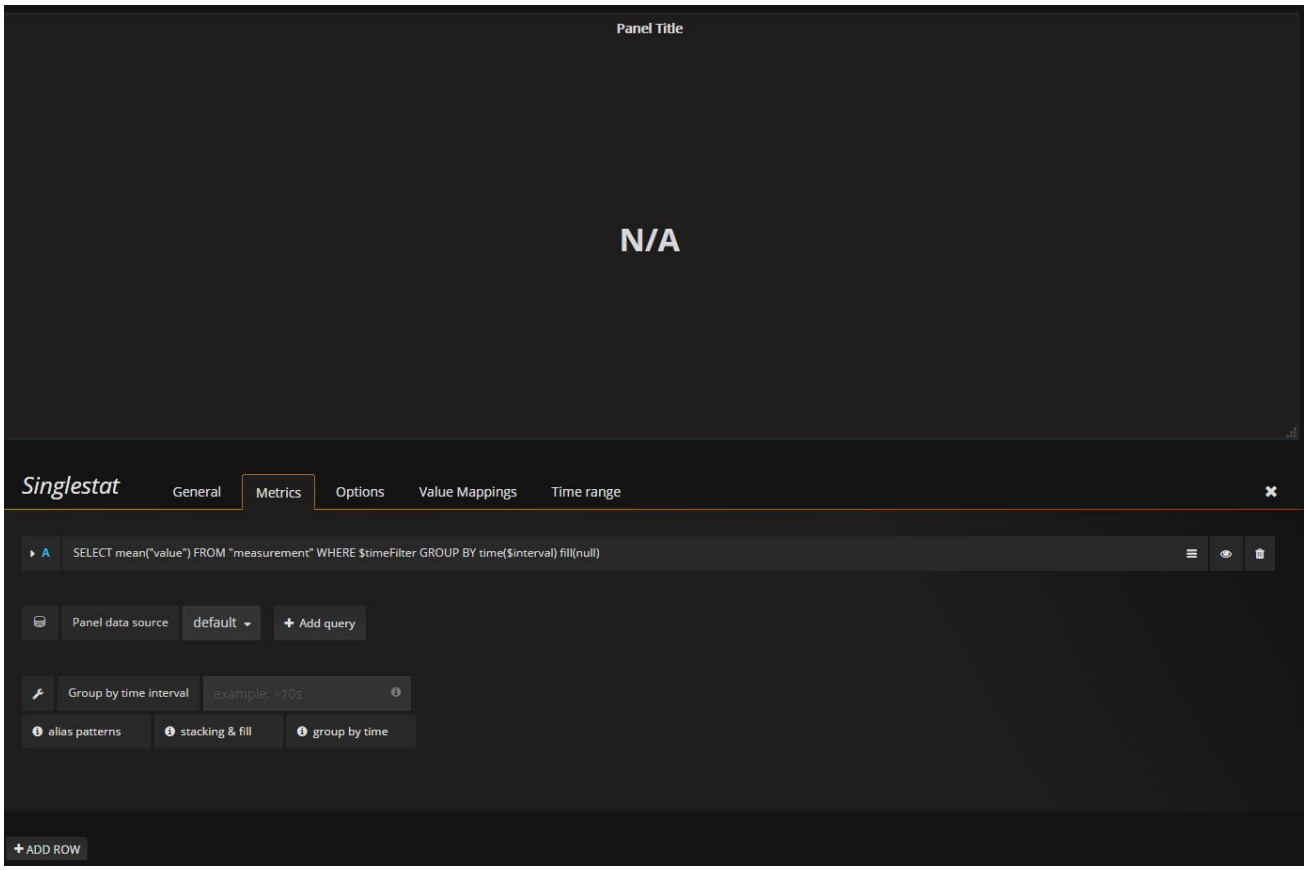


创建结果

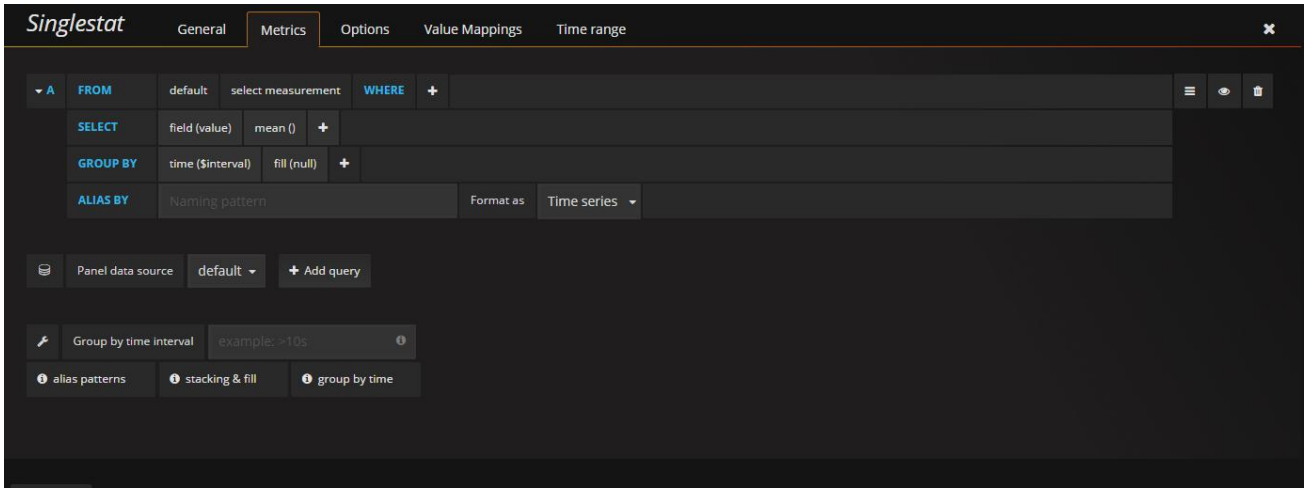


点击Panel Title----Edit新编辑一个面板的配置信息，点击还可以共享面板和查看面板具体信息

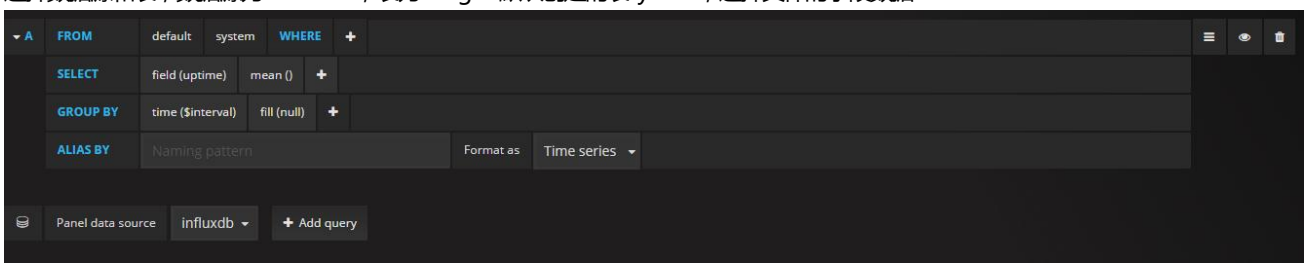
默认进入Metrics中进行配置，其中默认就有一条默认配置，该查询语句是根据您所选择的默认数据源类型来生成的默认语句。“A”表示一条语句，一个面板可以由多个查询组成，如果需要可点击Add query



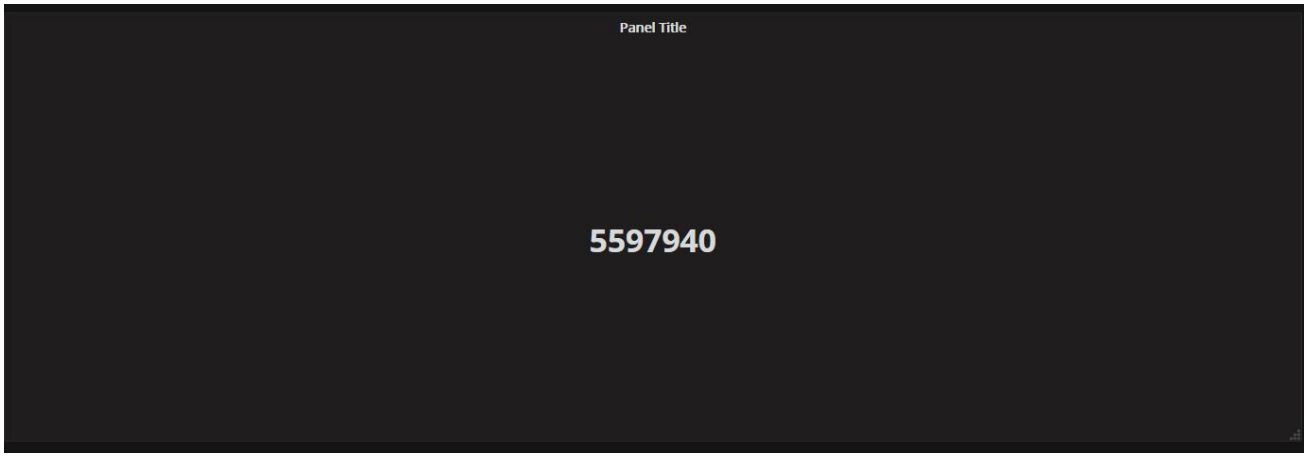
点击A查询展开查询



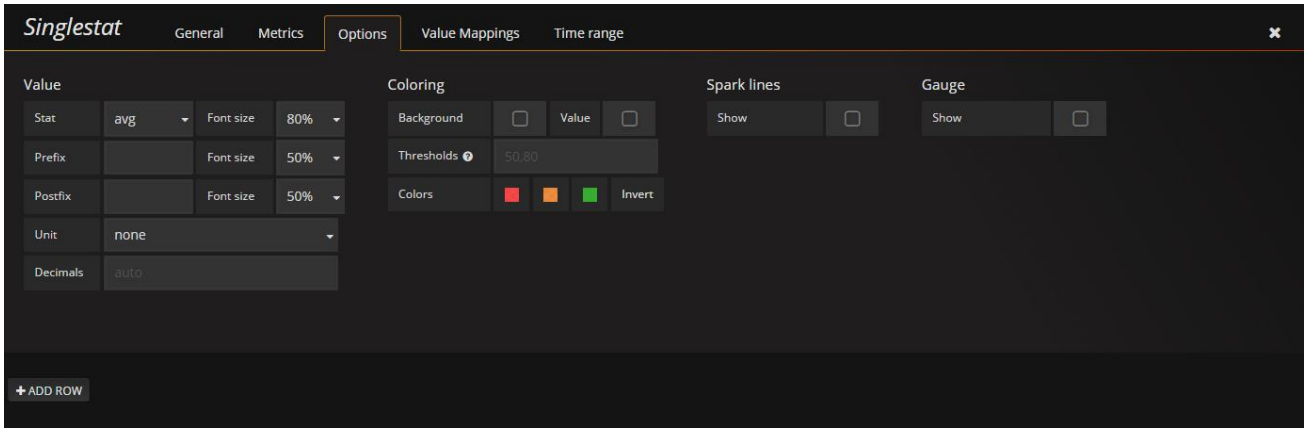
选择数据源和表，数据源为influxdb ，表为telegraf默认创建的表system，选择具体的字段数据



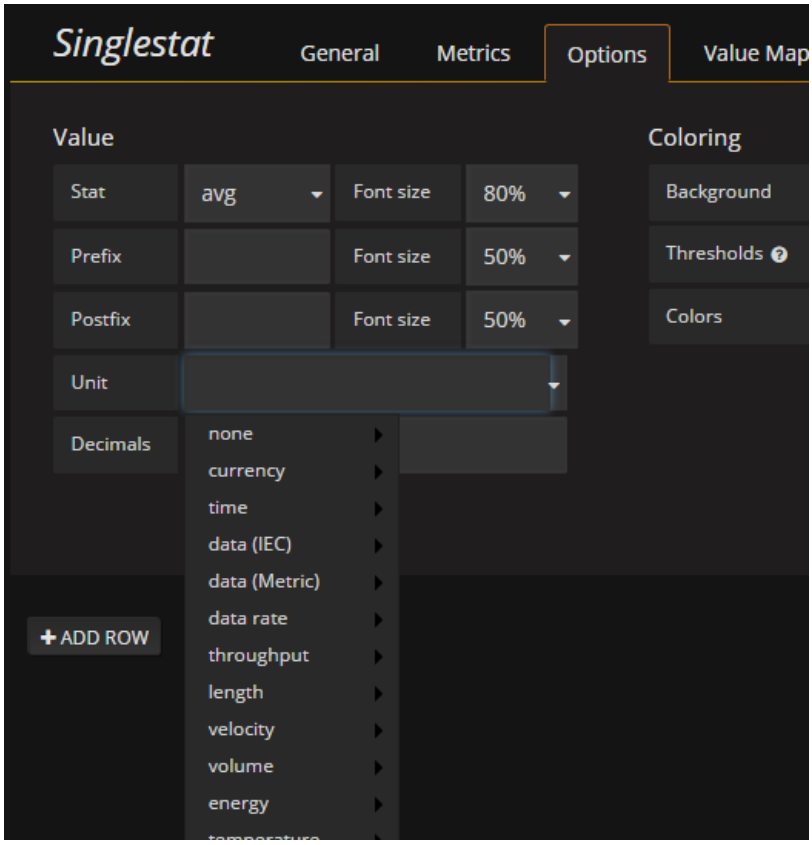
查询结果为5597940，默认为秒，那如何把数据转换成我们可以正常识别的值呢，点击Options

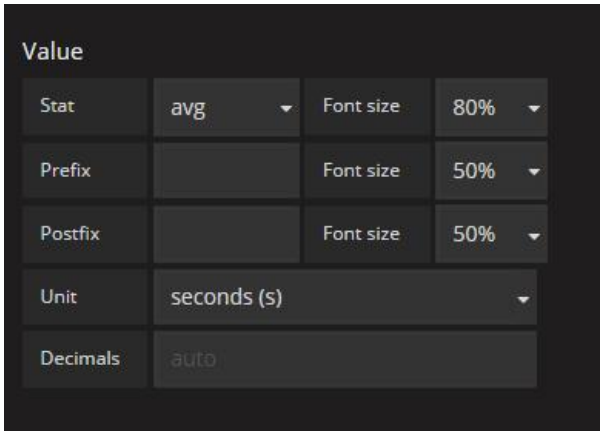


选择Unit单位

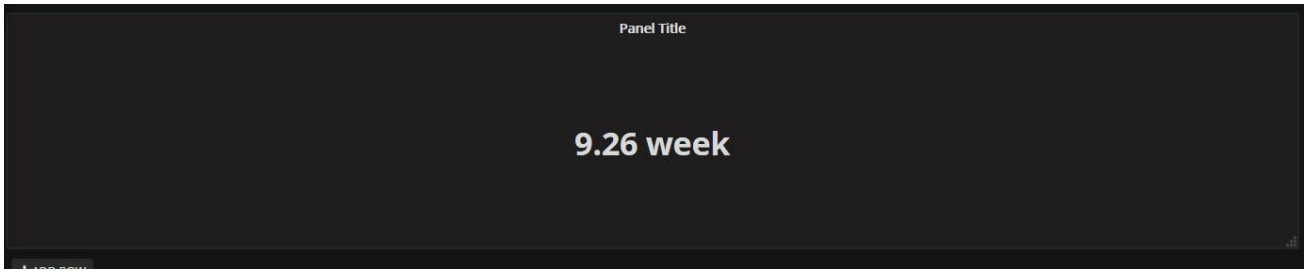


默认支持的单位格式有很多，我们选择secons(s)即可

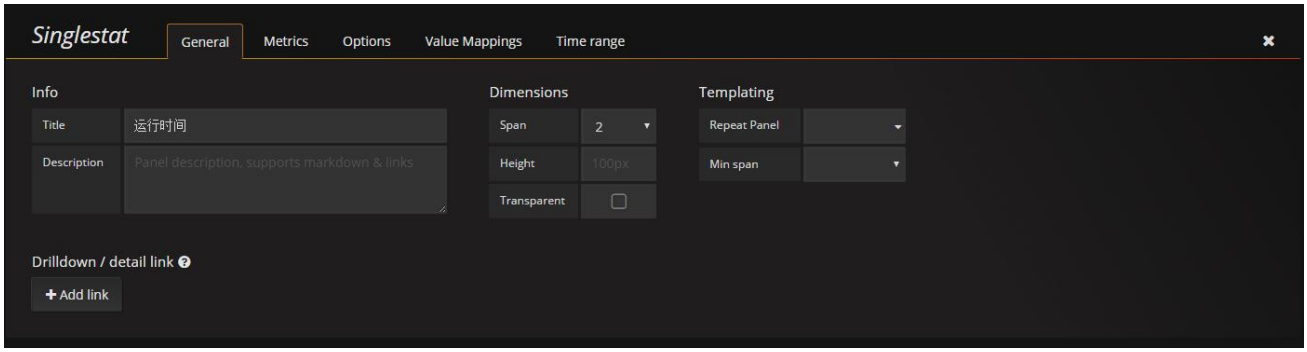




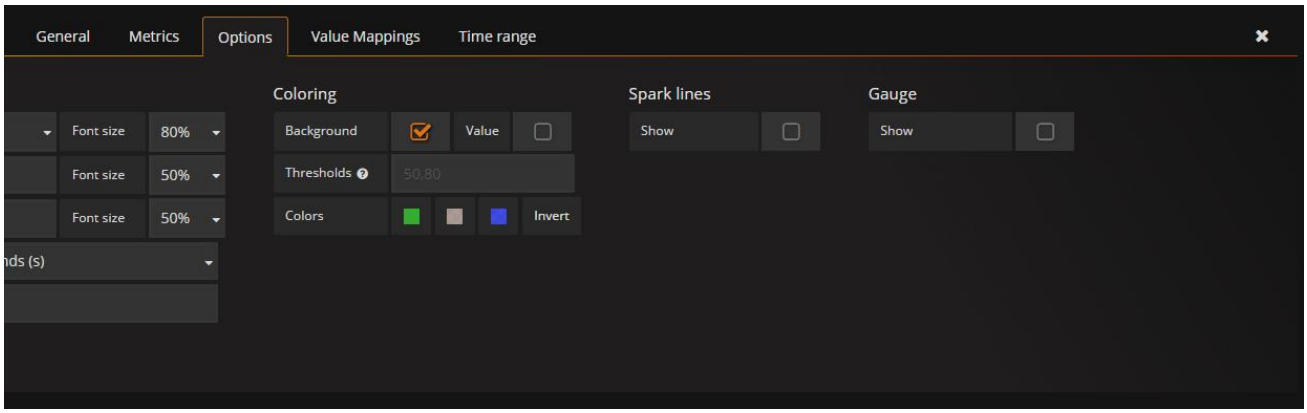
保存面板：Ctrl+s后点击x关闭后查看Dashboard



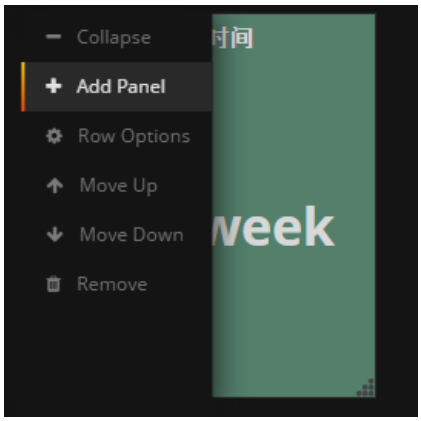
刚已经提到，一条Row默认有12行，我们此时需要调整span和title信息，ctrl+s保存



再次编辑面板选择Options--设置coloring--勾选backgroud--点击中间的colors选择背景颜色,也可以设置Thresholds的一个区间值，下面的colors就与此对应：比如50，80那第一个color为0-50的颜色标记，第二个color 为50-80，第三个为80-100的色值。

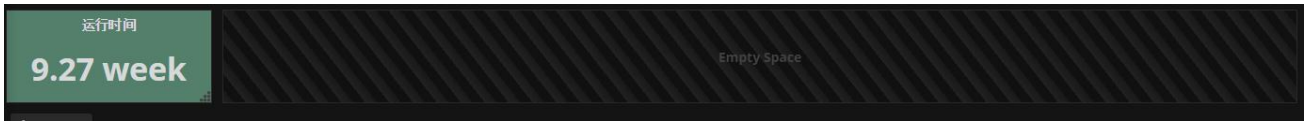


我们调整Row的高度，让它保存一个合理的高度--点击Row Options进行调整



调整高度值为100后保存ctrl+s,重新刷新浏览器后查看结果

那此时我们就建立好了一个面板

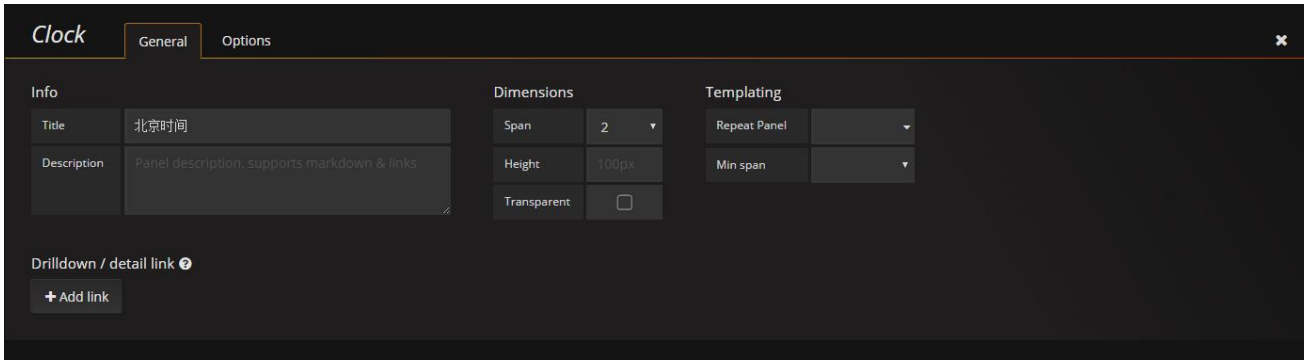


那后面以此建立即可

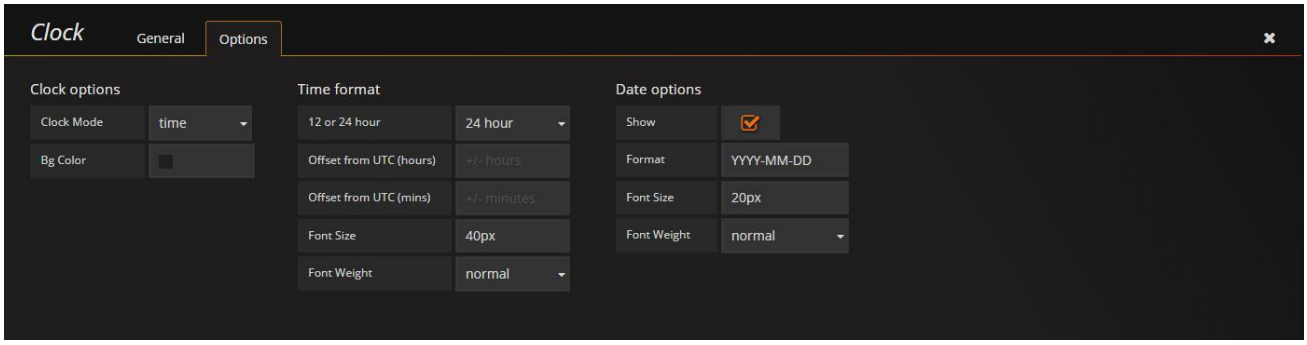
添加clock面板

点击Panel Title

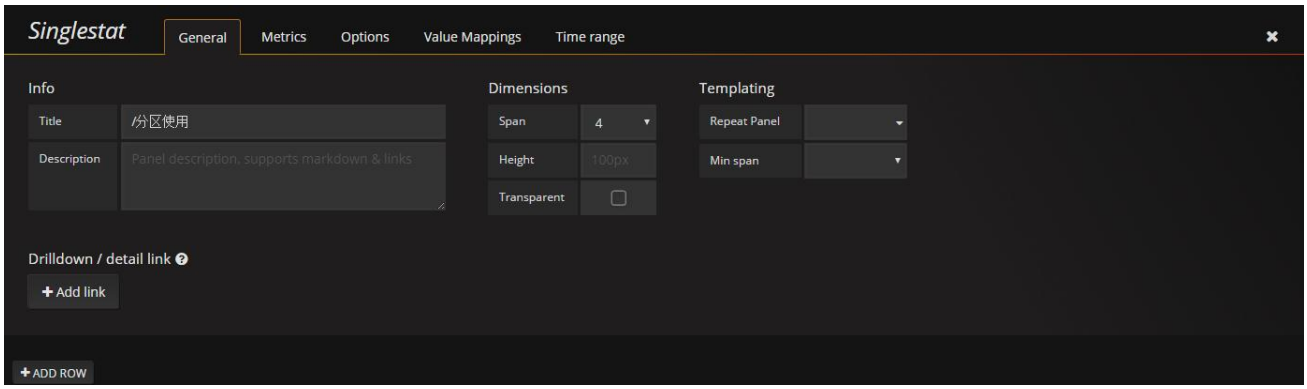
配置Title信息和span宽度

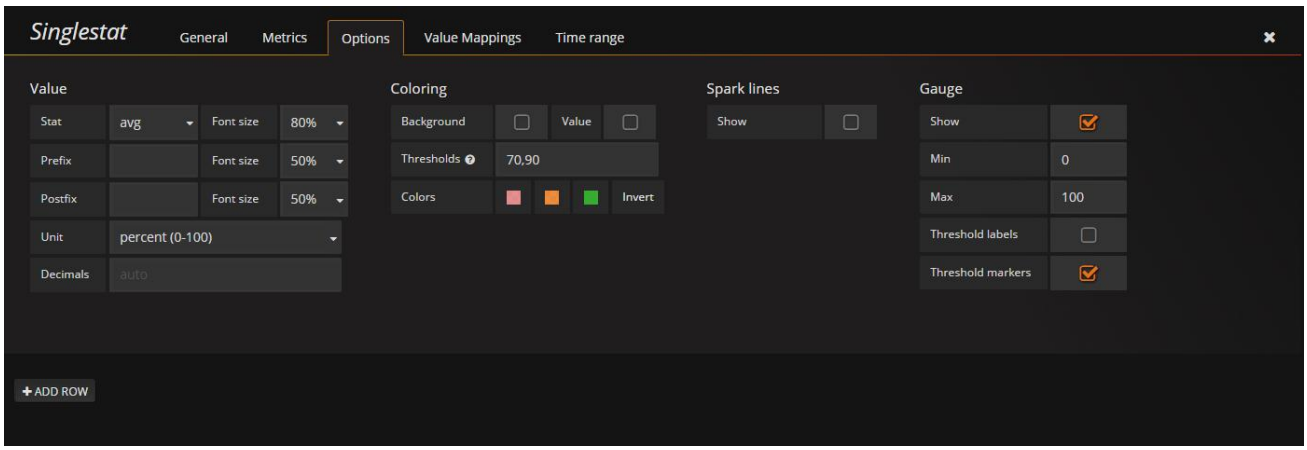


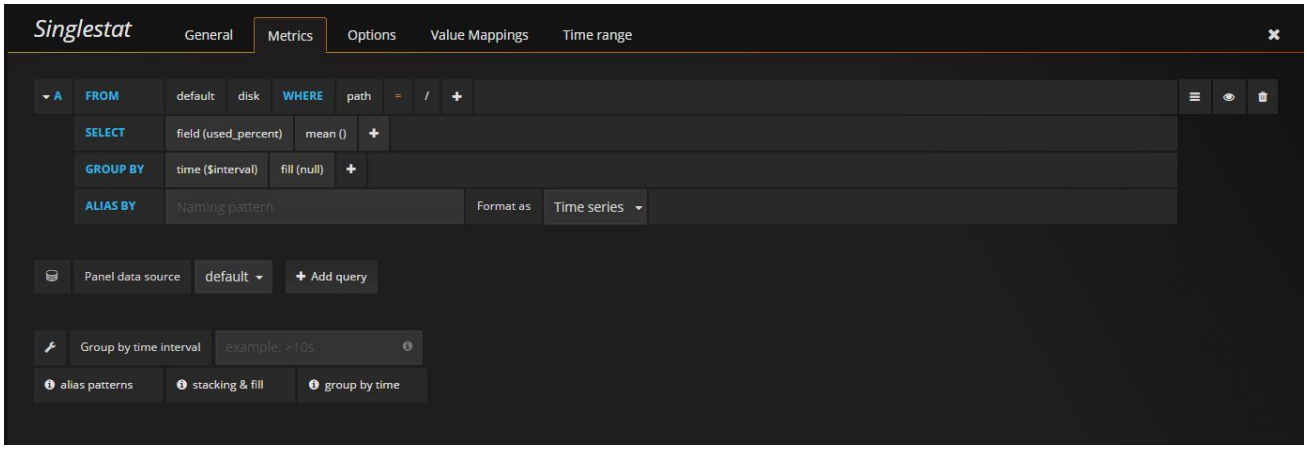
勾选Date Options显示格式，还可以自定义背景页面，调整字体大小为40px



配置/分区使用展示

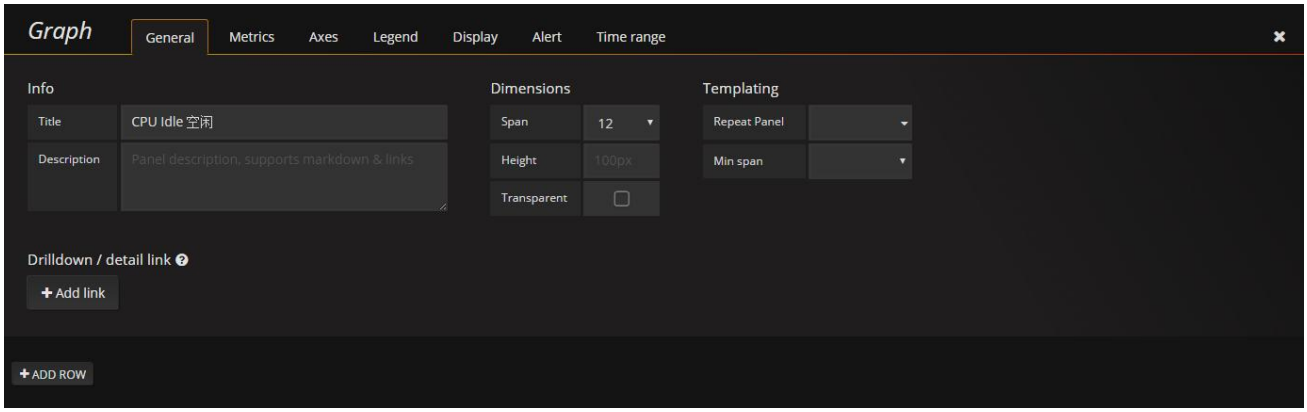


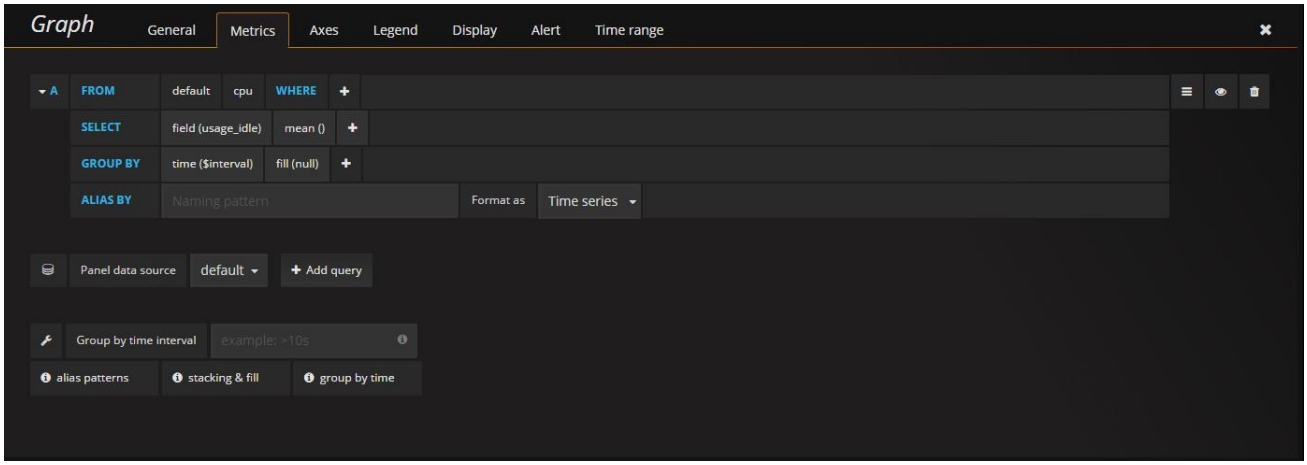




自定义绘图结果

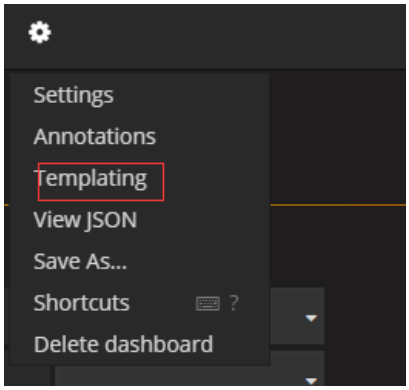
我们现在来定义一个CPU信息



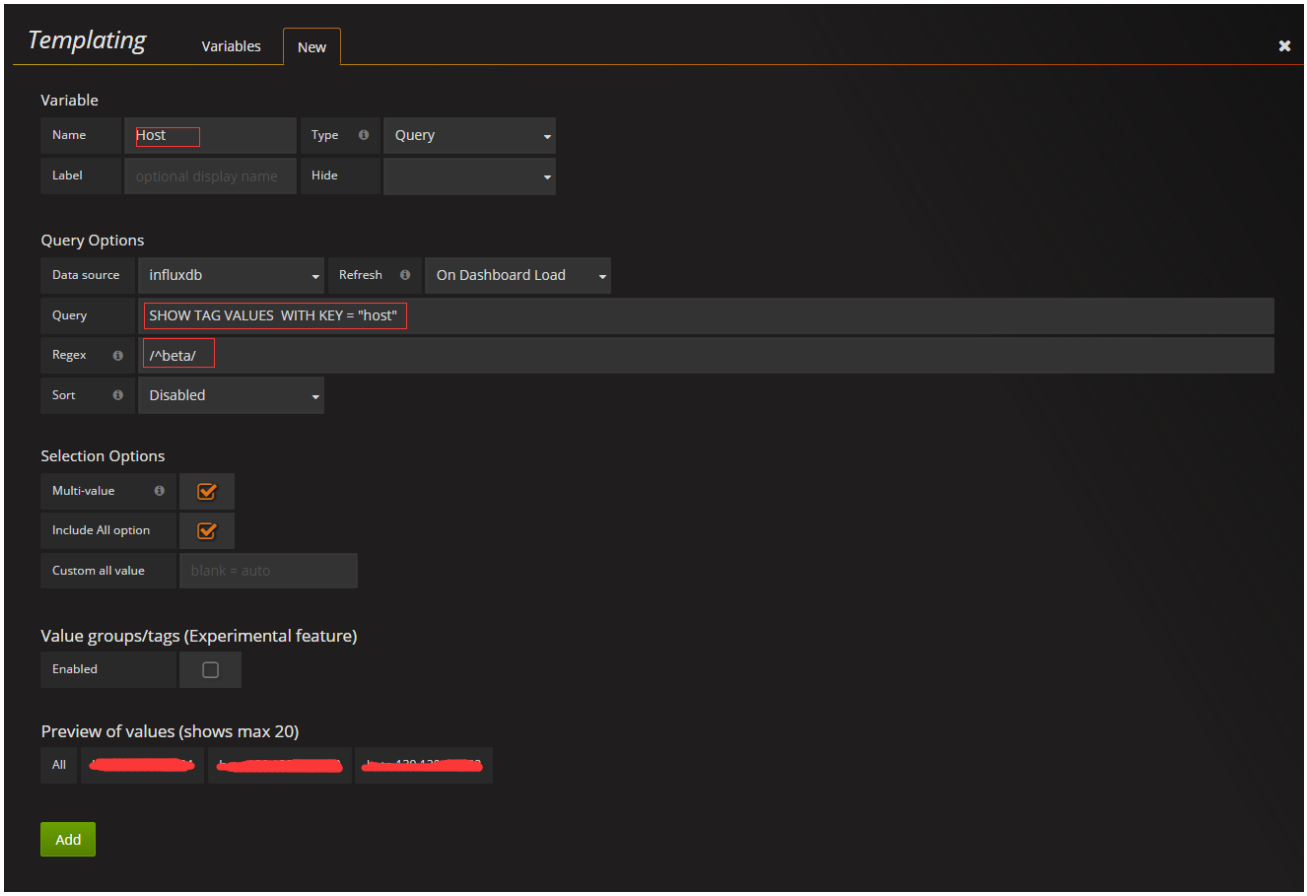


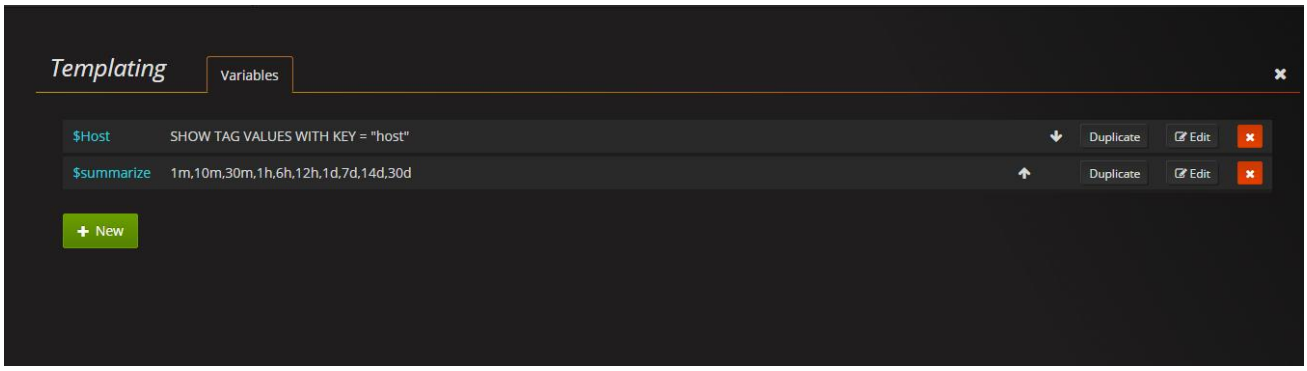
8.模板信息

可以把一个自定好的Dashboard制作成模板，那后面就可以直接使用该模板即可。如果一个模板中需要查看不同主机或者不同CPU，磁盘有情的监控信息。那此时就可以在模板中创建变量。

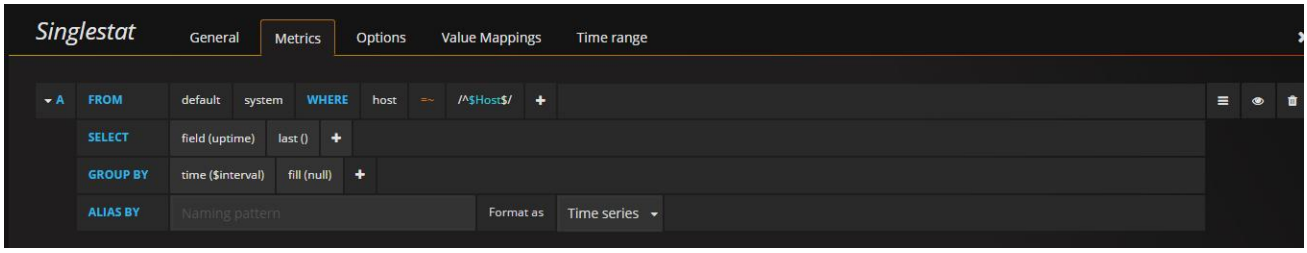


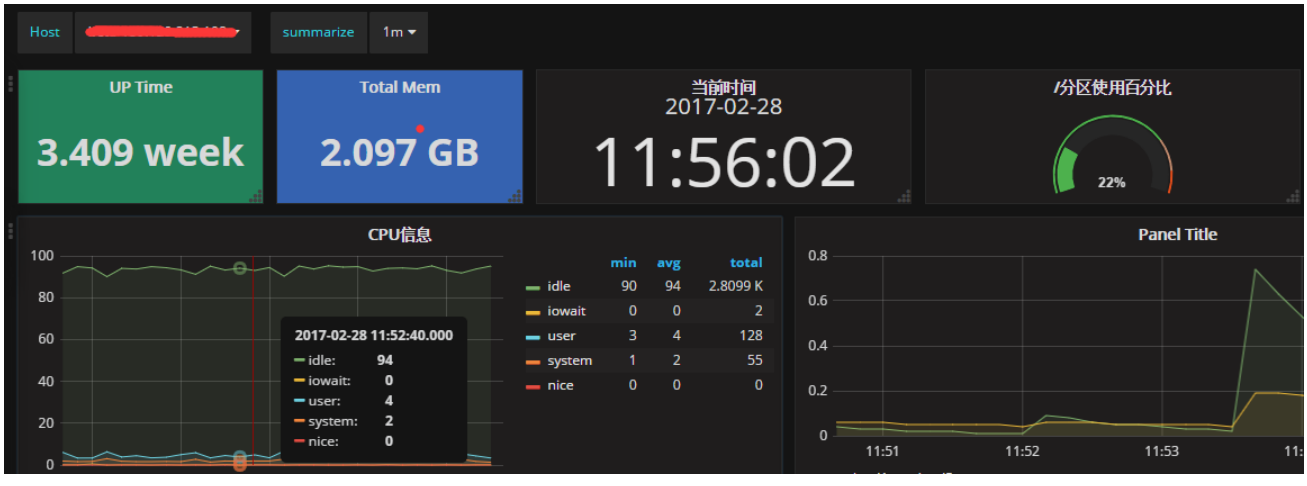
配置变量名为Host,数据来源为influxdb ,Query语句语法来源influxdb中，





调整面板中的metrics，加上where语句

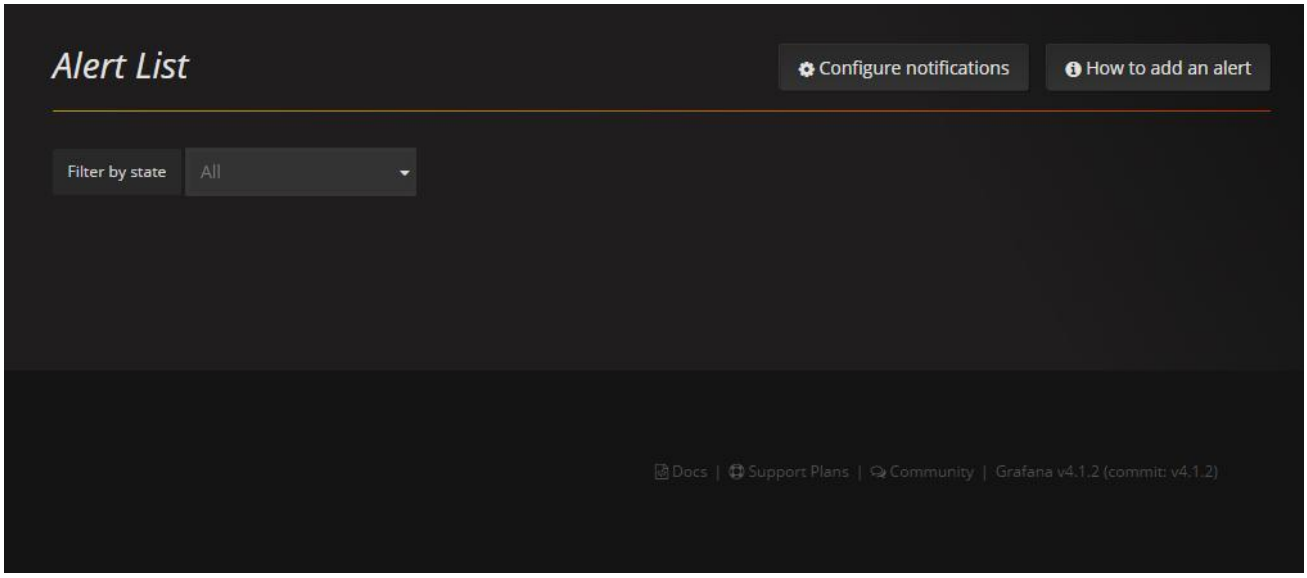


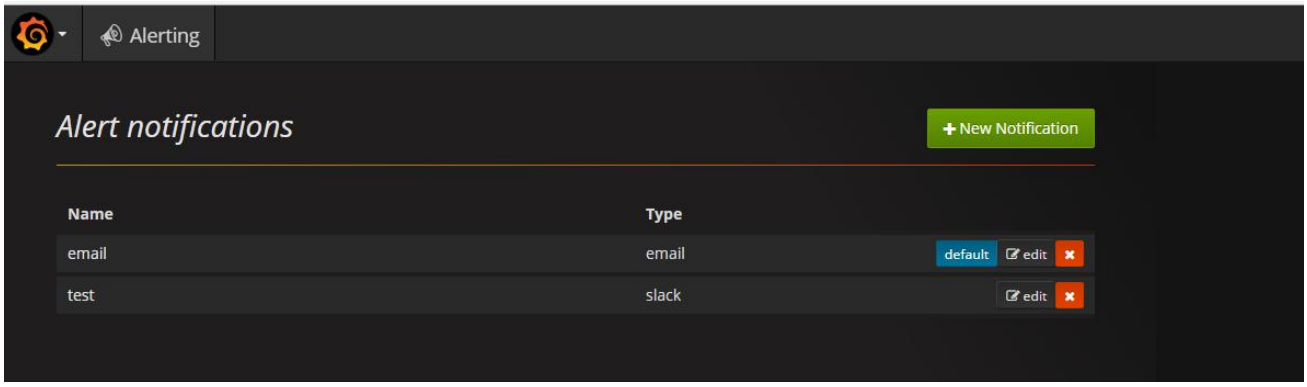


9.报警

grafana支持以下报警方式
1.email
2.slack
3.webhook
4.pagerduty
5.victorops
6.opsgenie

配置报警
1.需要添加报警 notifications



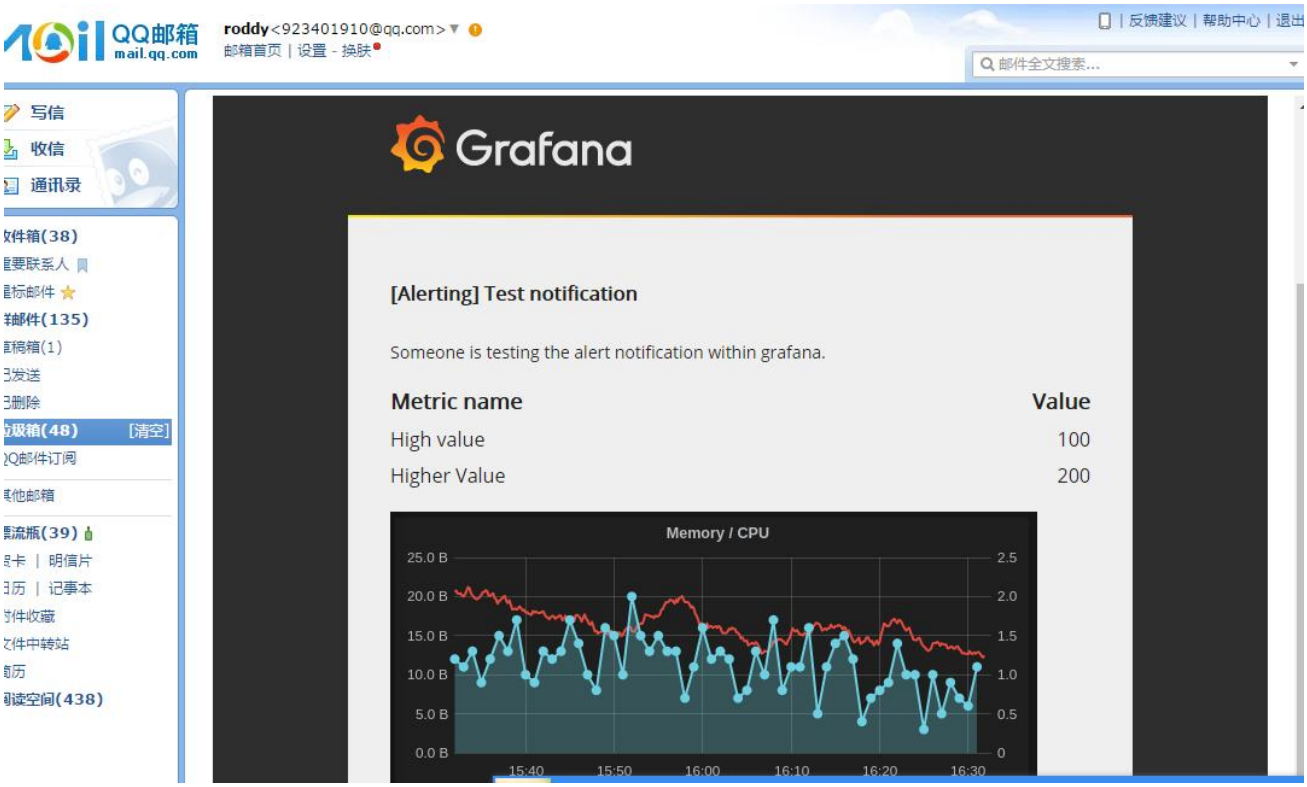


2.修改smtp配置

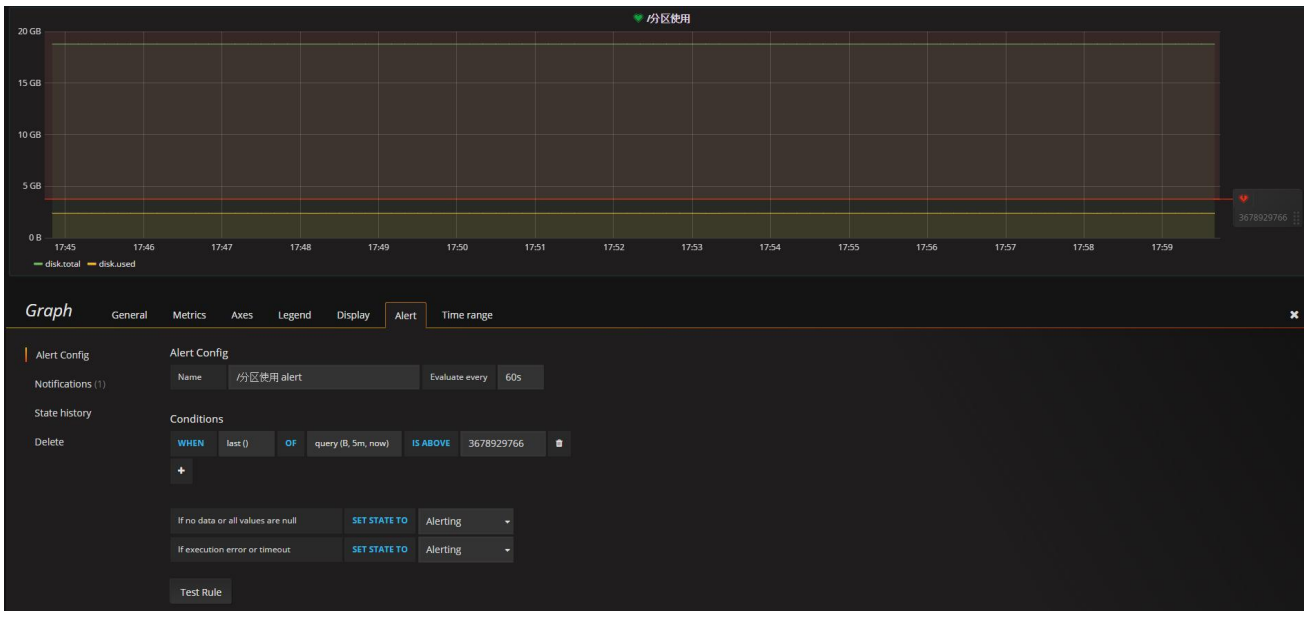
```
[smtp]
enabled = true
host = smtp.sina.com:25   #必须加端口
user =  roddy1219@sina.com
password =  xxxxxxxx
;cert_file =
;key_file =
skip_verify = true
from_address = roddy1219@sina.com
```

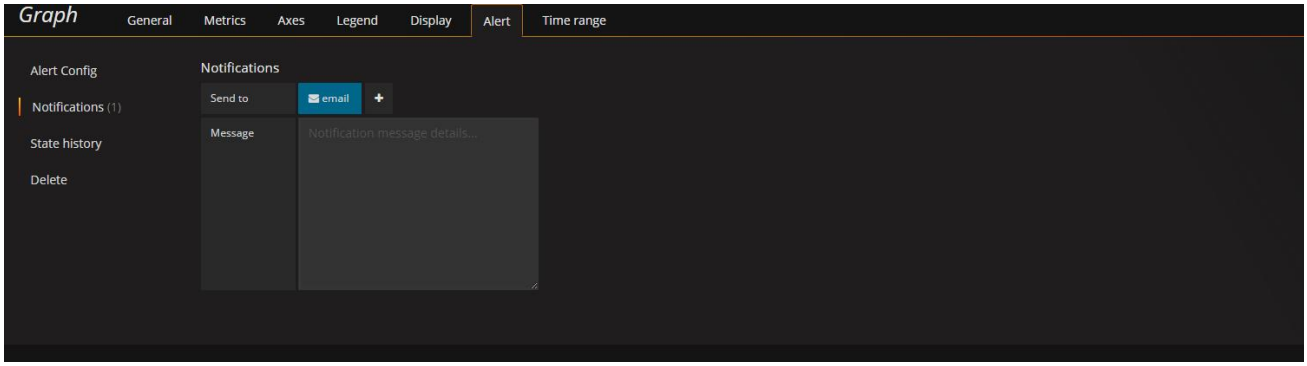
[alerting]
# Makes it possible to turn off alert rule execution.
execute_alerts = true

3.测试



5.配置报警





可以看到此监控的状态改变情况