Grafana配置与简单使用

[liangdong@smzdm.com](mailto:liangdong@smzdm.com)

[安装 1](#_Toc534383108)

[配置文件 2](#_Toc534383109)

[database数据库 2](#_Toc534383110)

[Session登录会话 2](#_Toc534383111)

[Security安全 3](#_Toc534383112)

[认证系统 3](#_Toc534383113)

[GUI配置 4](#_Toc534383114)

[配置数据源 4](#_Toc534383115)

[查看用户user 6](#_Toc534383116)

[查看组织org 7](#_Toc534383117)

[创建team 9](#_Toc534383118)

[创建folder 11](#_Toc534383119)

[创建dashboard 12](#_Toc534383120)

[创建pannel 13](#_Toc534383121)

[配置dashboard的权限 15](#_Toc534383122)

[模板变量 17](#_Toc534383123)

[自动刷新 20](#_Toc534383124)

[完 20](#_Toc534383125)

下面简单记录一下grafana的使用流程。

# 安装

打开下载页：<https://grafana.com/grafana/download?platform=mac>

选择对应的平台，走wget下载二进制即可，别着急启动。

# 配置文件

解压后，conf目录下defaults.ini是默认配置文件，sample.ini是自定义配置，grafana启动时候需要-config指定加载sample.ini文件，并且会自动load同目录下的defaults.ini作为默认参数，千万不能删除defaults.ini，否则无法启动grafana。

ini 配置可以参考官方文档学习，比较精炼：<http://docs.grafana.org/installation/configuration/> 。

## database数据库

我改了一下数据库，grafana所有的配置都会存在数据库里，我用mysql：

[database]

# You can configure the database connection by specifying type, host, name, user and password

# as separate properties or as on string using the url properties.

# Either "mysql", "postgres" or "sqlite3", it's your choice

type = mysql

host = 127.0.0.1:3306

name = grafana

user = root

# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""

password = xxxxxyyyyy

注意，数据库提前建好：

create database grafana

数据表的话grafana会自动创建出来，Golang的ORM库大多这个风格。

## Session登录会话

为了可以部署grafana集群，需要让session存在redis里共享：

[session]

# Either "memory", "file", "redis", "mysql", "postgres", default is "file"

provider = redis

# Provider config options

# memory: not have any config yet

# file: session dir path, is relative to grafana data\_path

# redis: config like redis server e.g. `addr=127.0.0.1:6379,pool\_size=100,db=grafana`

# mysql: go-sql-driver/mysql dsn config string, e.g. `user:password@tcp(127.0.0.1:3306)/database\_name`

# postgres: user=a password=b host=localhost port=5432 dbname=c sslmode=disable

provider\_config = `addr=127.0.0.1:6379,pool\_size=100,db=grafana`

简单配置一下就行。

## Security安全

[security]

# default admin user, created on startup

admin\_user = admin

# default admin password, can be changed before first start of grafana, or in profile settings

admin\_password = admin

默认管理员是admin,admin，也可以登录后修改，届时帐号密码就保存到mysql里了。

## 认证系统

支持基于数据库的用户系统，或者LDAP，以及github等第三方登录。

默认走数据库。

Ldap相关的：

#################################### Auth LDAP ##########################

[auth.ldap]

;enabled = false

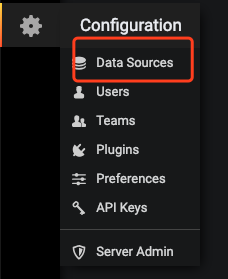
;config\_file = /etc/grafana/ldap.toml

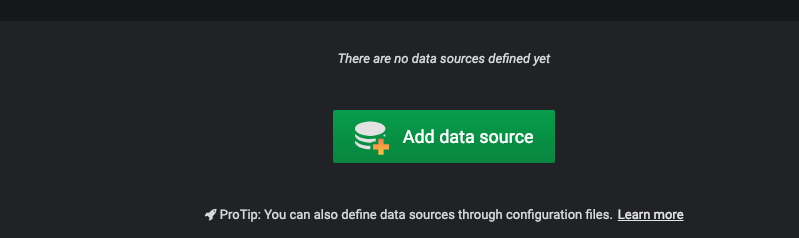
;allow\_sign\_up = true

# GUI配置

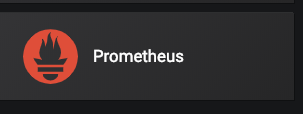
## 配置数据源

Grafana需要拉取prometheus的接口获取数据，才能画图。

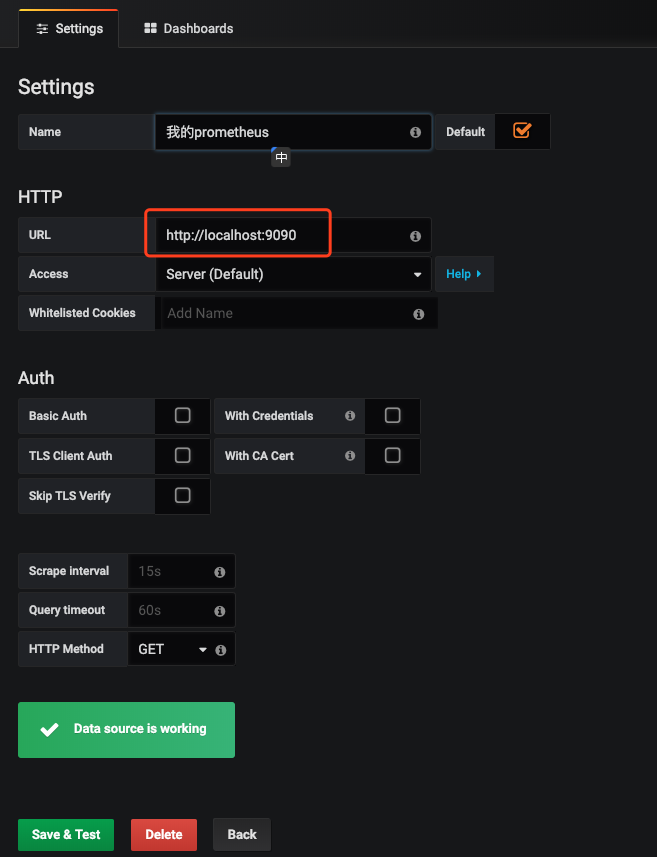


然后添加datasource：

选择prometheus：

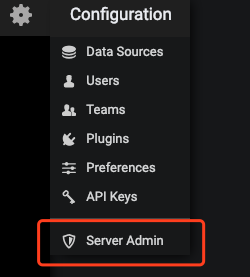


配置服务端地址：

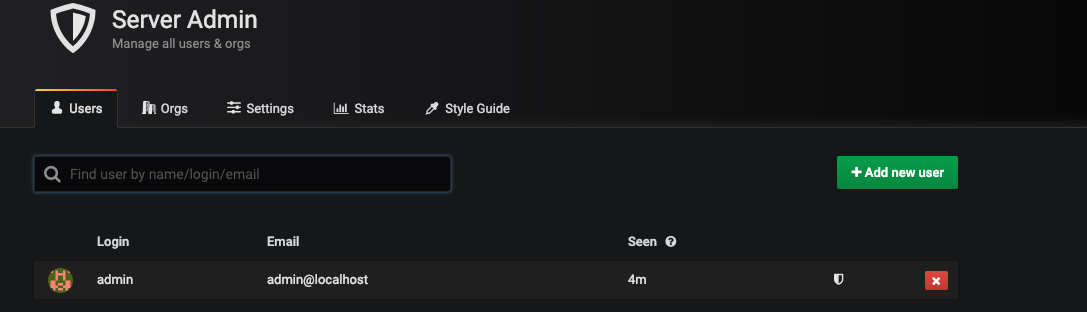


这样就OK了。

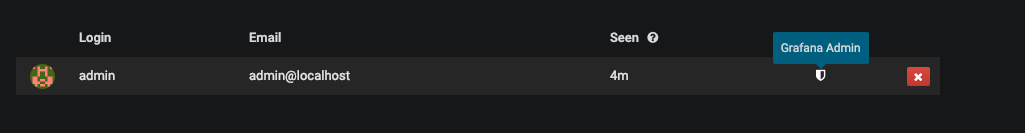
# 查看用户user



点击user，可以看到admin用户：

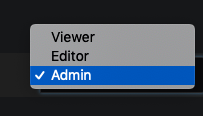


每个用户有一个角色：



Grafana一共就3种角色。

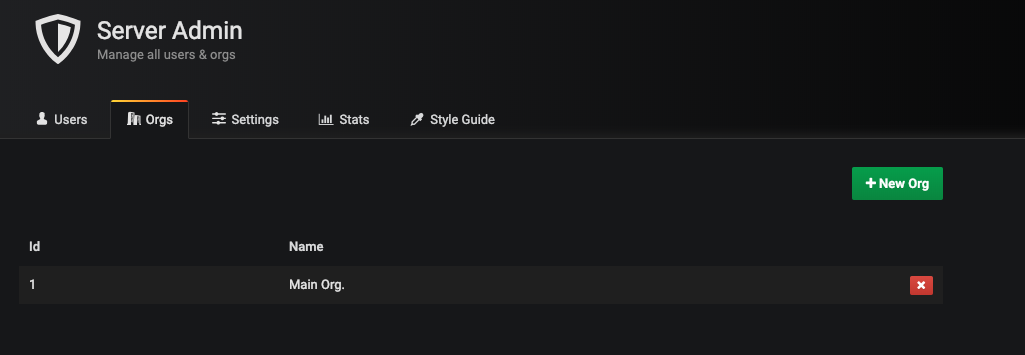
admin用户的角色叫做admin，另外2个角色叫做viewer和editor，过会说。



## 查看组织org

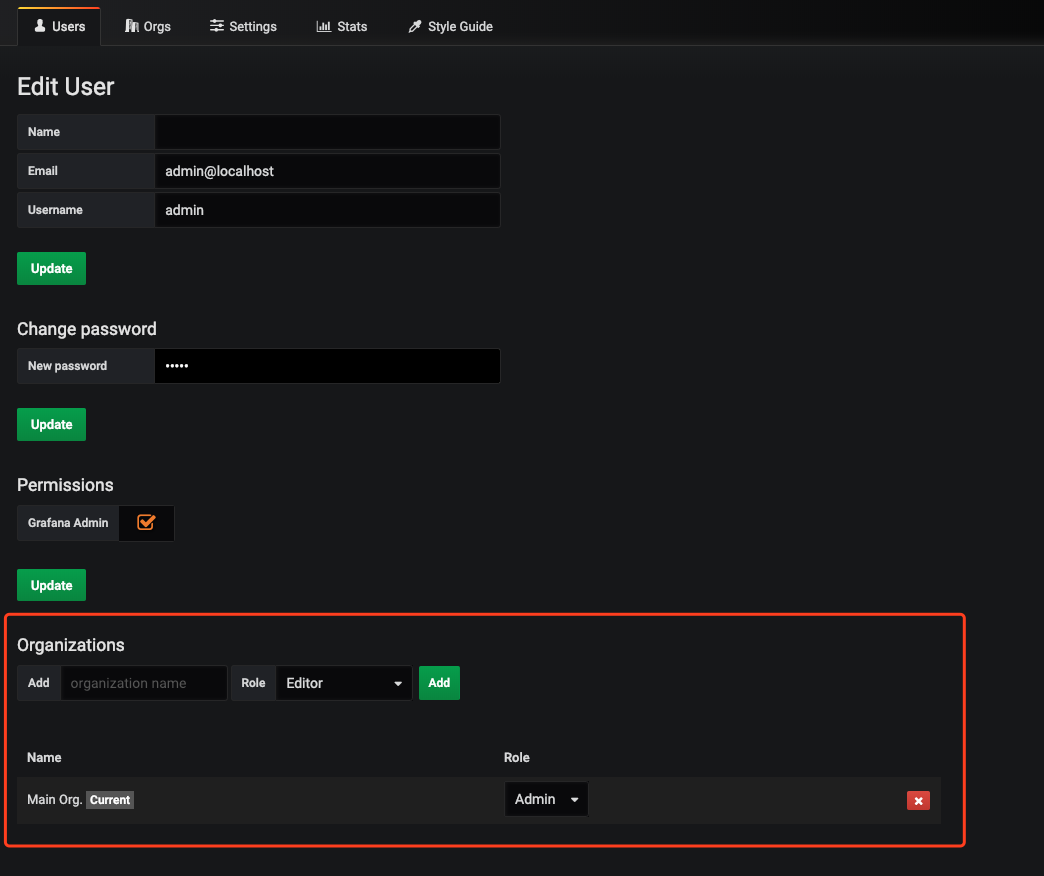
Org相当于给各个业务线划分一下命名空间，PHP团队的空间下放着PHP团队的各种dashboard，JAVA团队的放着JAVA的dashboard。

grafana默认就一个org：

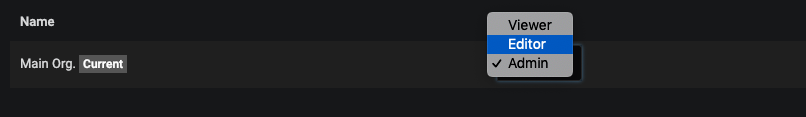


user与org的关系是1:N，比如大BOSS可以看PHP和JAVA的dashboard，所以属于php和java两个org。

点user可以查看user详情：

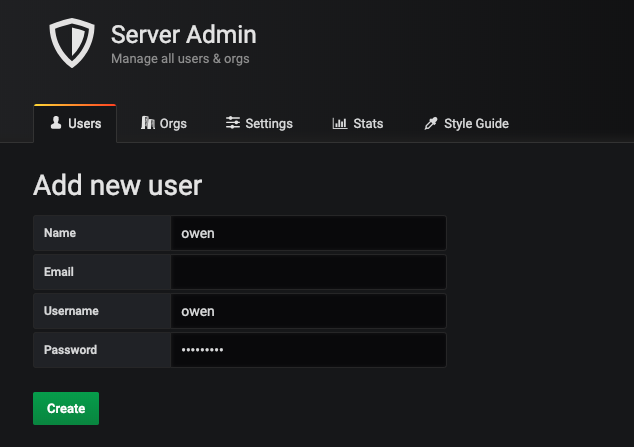


注意到，最下面就是让admin用户加入到N个org里，在每个Org里扮演某个role：

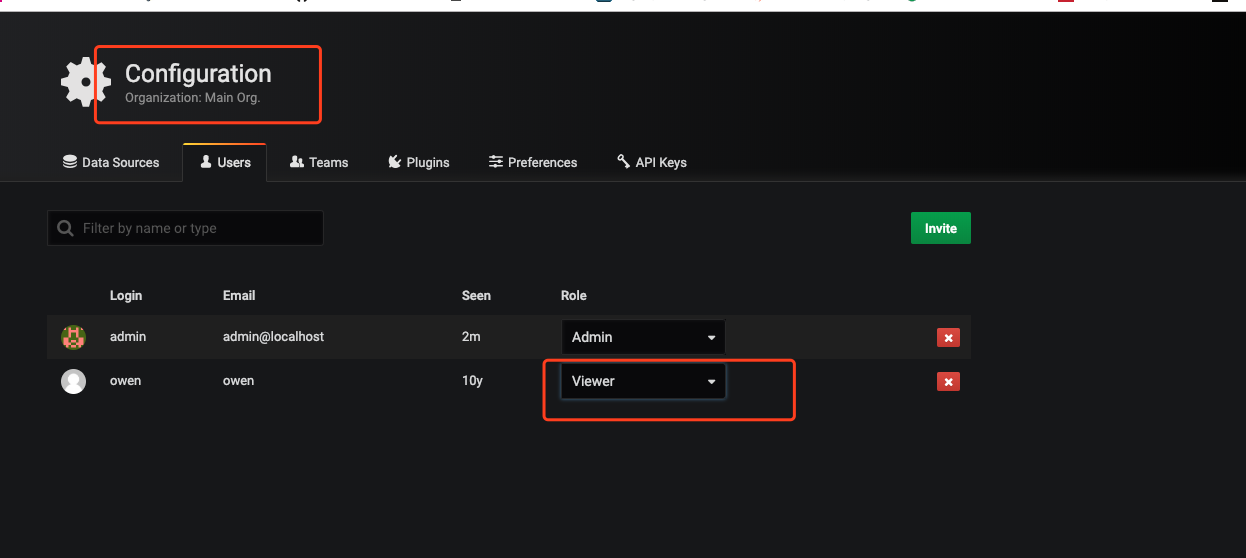


这里admin用户在Main org扮演admin这个角色。

我可以创建一个新用户：



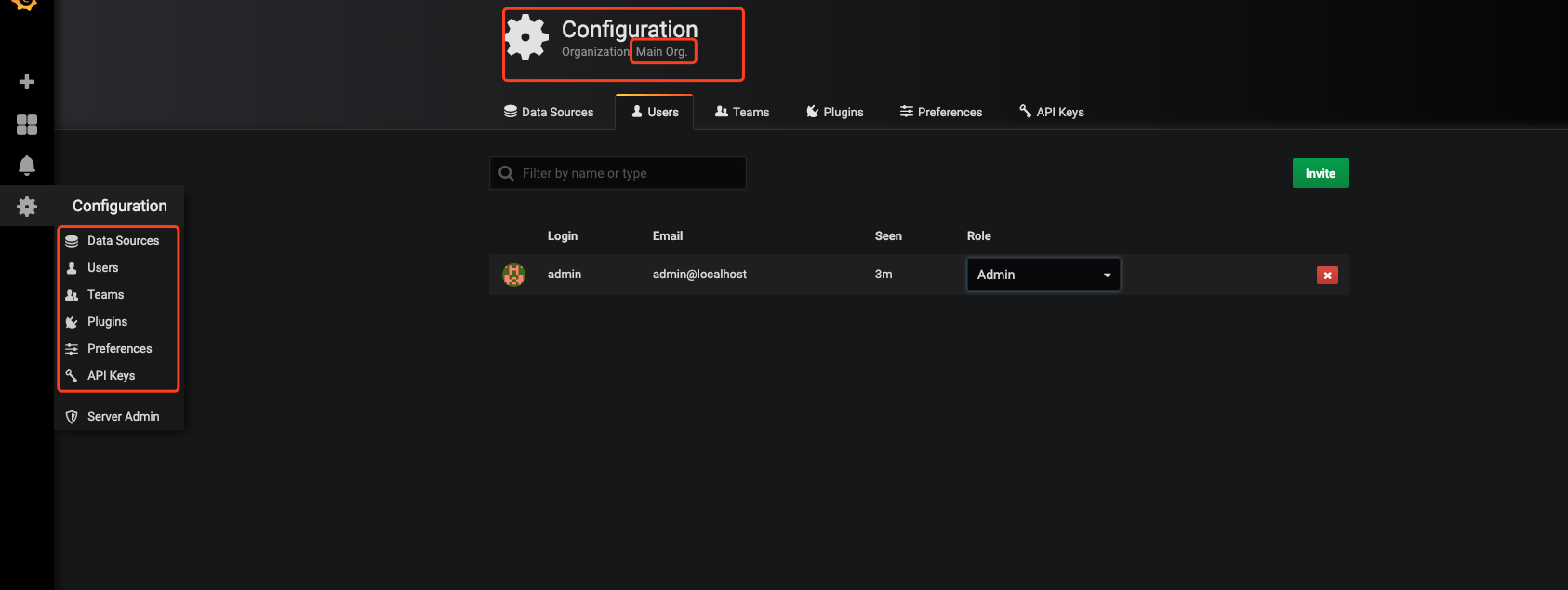
User在每个org下可以只指定1个role：



记住：user在某个org下，只有一个role； user在不同org下的role可以各不相同。

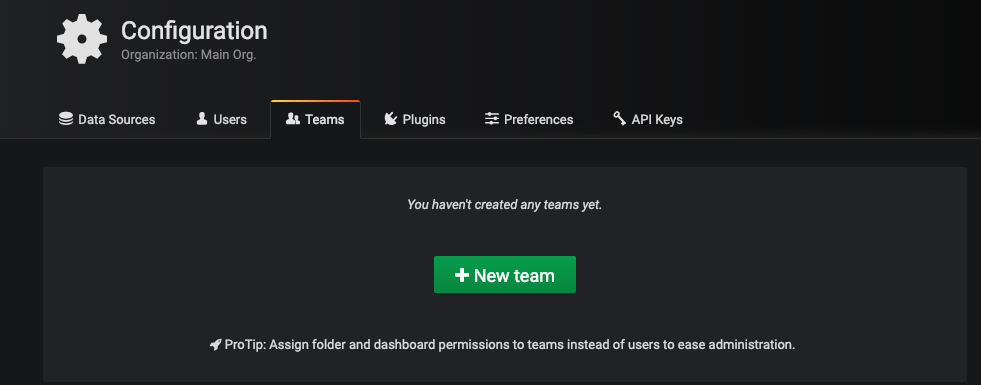
## 创建team

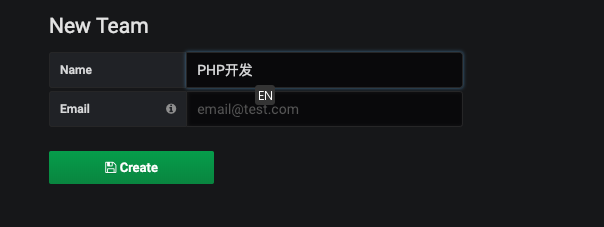
除了server admin按钮，下面的这些按钮：



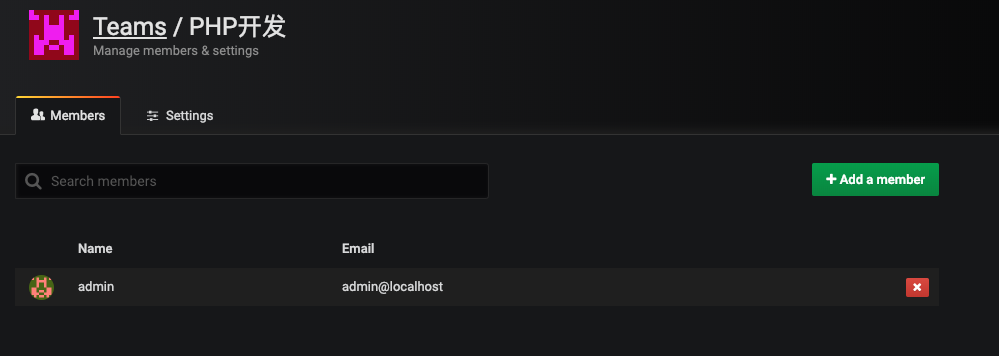
都是针对具体某个org之下的配置，只有server admin按钮是全局配置。

我们可以在某个org下，创建team代表一类用户：





给team添加org下的成员：

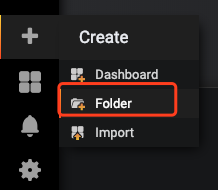


Team一会会演示有什么用。

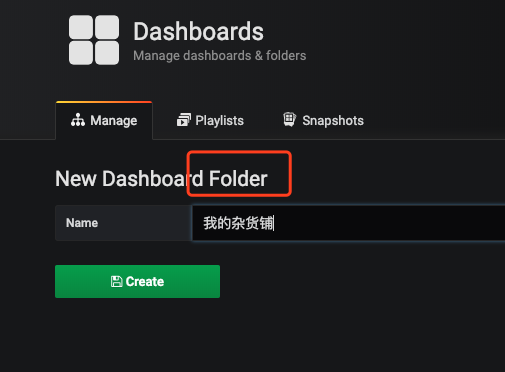
## 创建folder

可以在org下创建folder。

什么是folder呢？就是一个目录，里面放一堆dashboards。

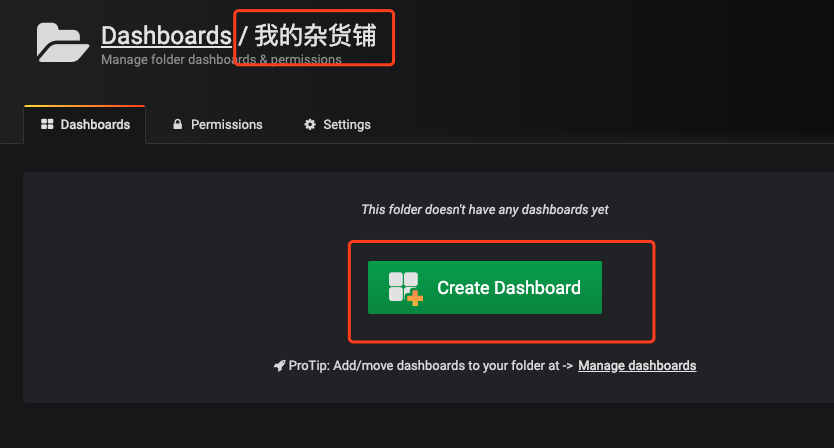


创建一下folder，从标题可以看出folder就是dashboard的集合：



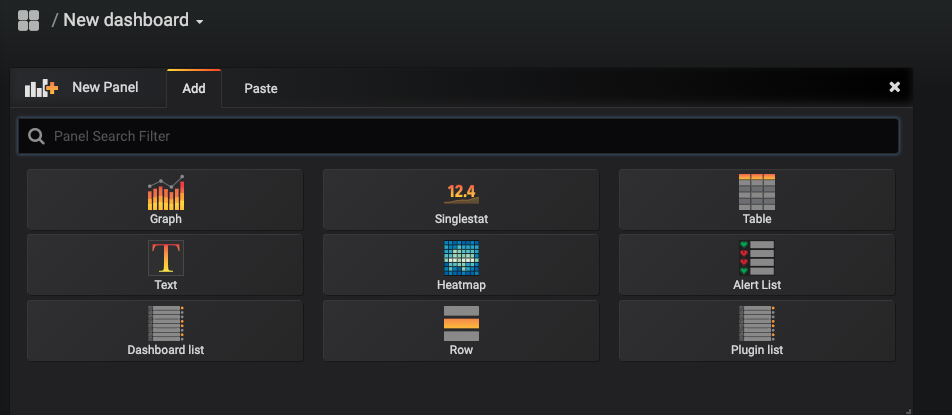
## 创建dashboard

接下来就是在这个folder内创建dashboard：

  
dashboard其实就是一个空页面，里面需要添加一个一个的图表，也叫做panel。

## 创建pannel

刚创建dashboard的时候，会提示你创建第一个panel：

一般都是选graph panel，画点曲线啥的。

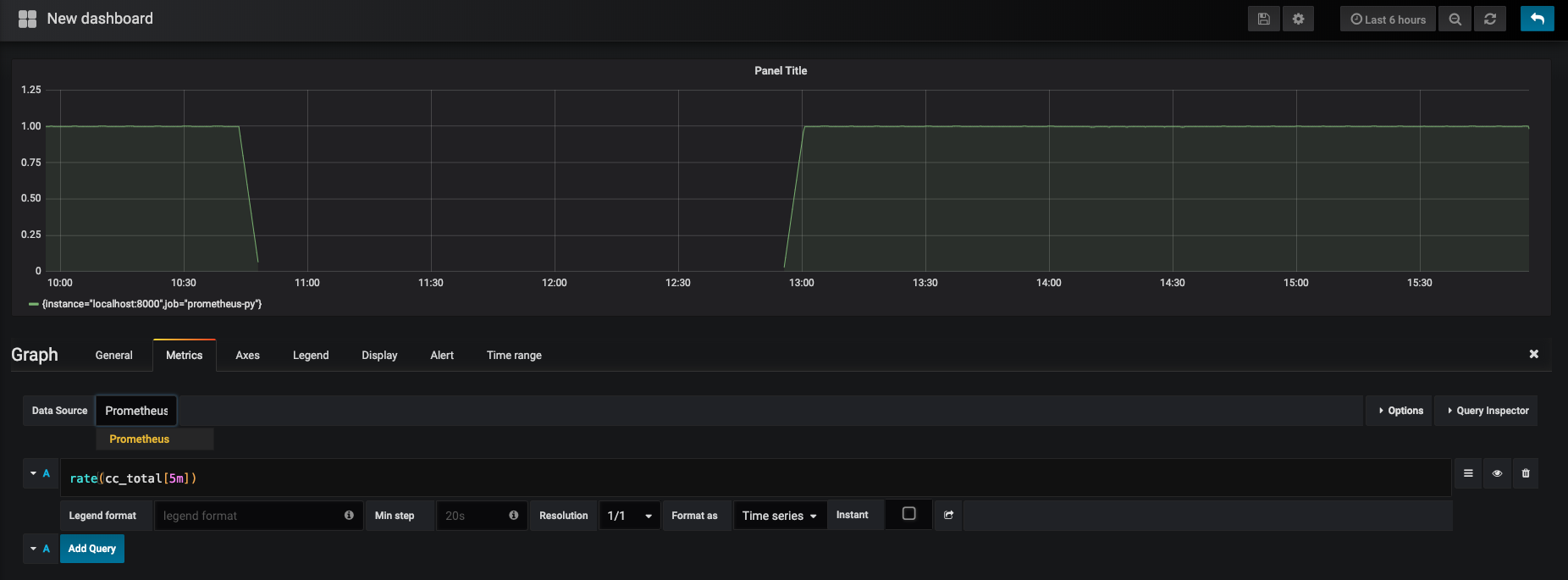


创建的graph没有配置数据来源，我们可以配置一下：

点edit：



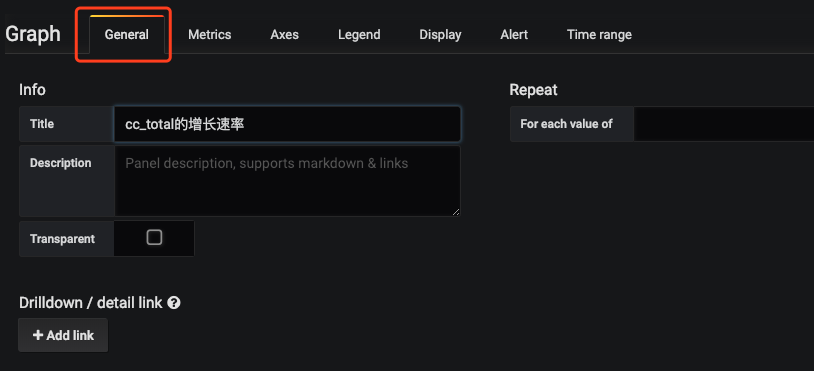
选择配置过的prometheus datasource：



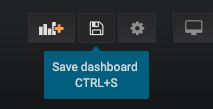
添加promql查询计算cc\_total的增长速率，然后就出现了曲线。

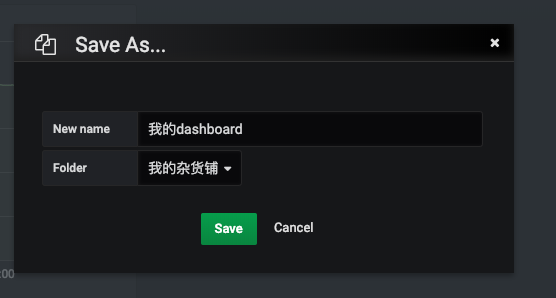
Cc\_total是python客户端每隔1秒inc一次的，被prometheus采集走的，所以计算的rate基本保持在1，即每秒增加1。

可以给panel起个名字：



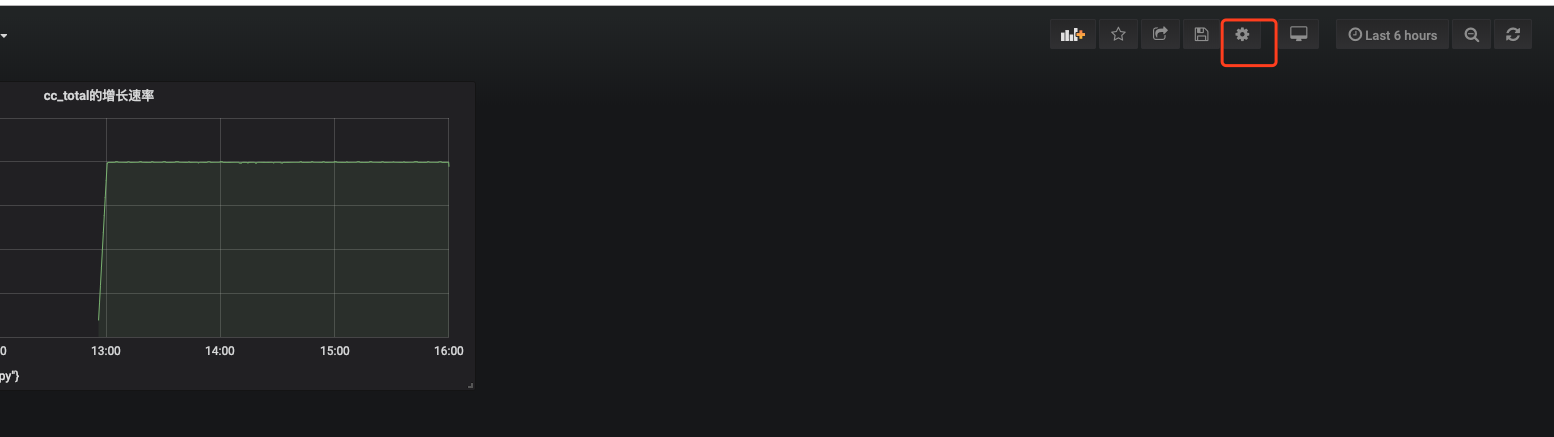
随便拖拽大小：

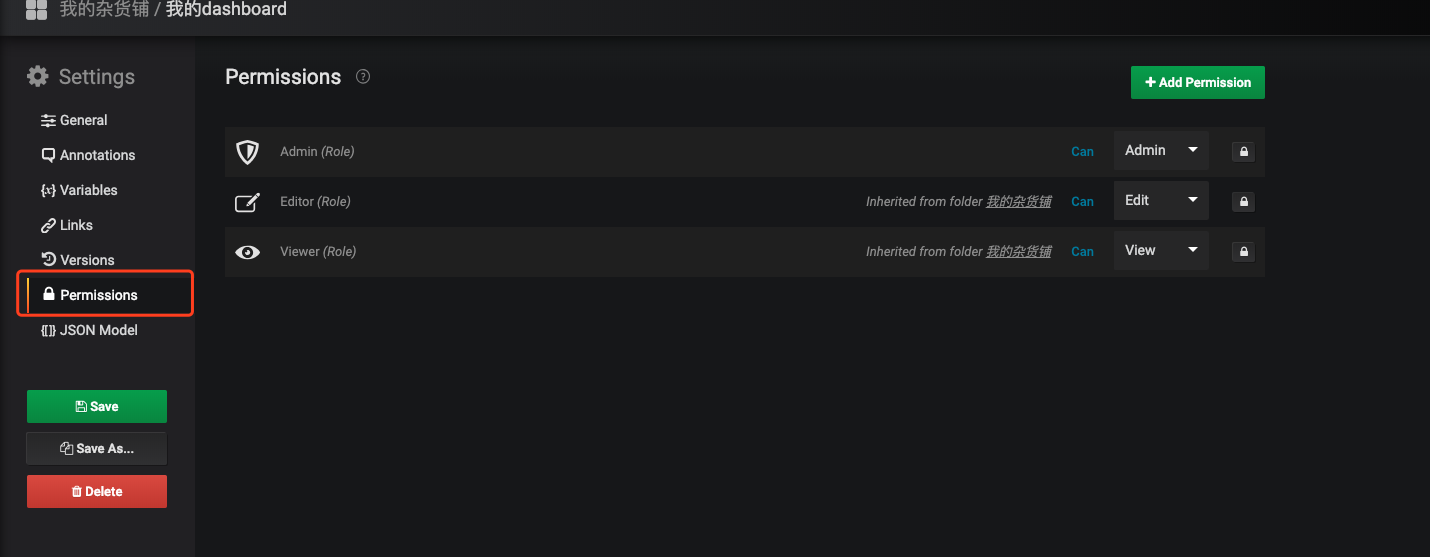
 

屏幕右上角点save保存整个dashboard的所有panel到数据库，记得选择folder：

## 配置dashboard的权限

点击dashboard编辑：





这里已经有3条权限配置，都是针对user的配置。

意思就是：

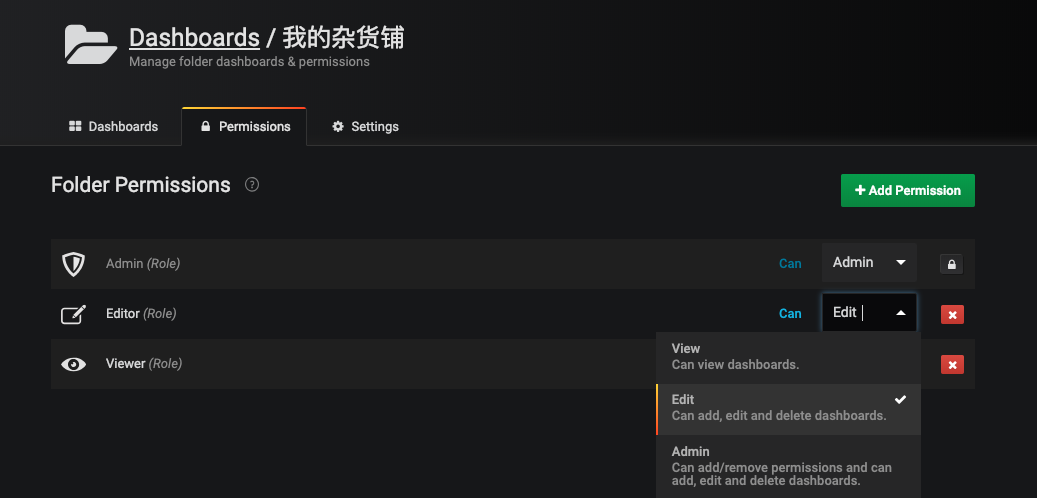
1，如果user在该org下role=admin，就拥有了对该dashboard的admin权利，爱干啥干啥，比如配置这个dashboard，看dashboard，配置dashboard的权限。

2，如果user在该org下role=**editor**，就拥有了对该dashbaord的**edit**操作权限，可以编辑dashboard，当然也可以看。

3，如果user在该org下role=**viewer**，就拥有对该dashbaord的**view**操作权限，就是可以看这个dashboard，但不能编辑。

注意editor是user的role，edit是action或者说权限。

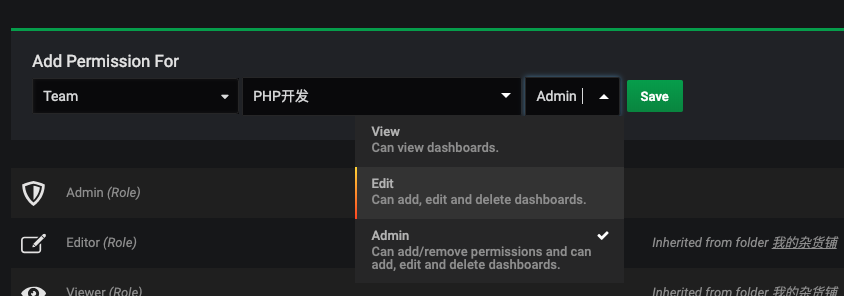
这3个dashboard的权限配置，是继承自folder『我的杂货铺』的：



可见，该org下哪个role对该folder有什么操作权限（admin/edit/view），会继承到folder下的dashboard，在dashboard中无法覆盖。

除了这种user role -> admin/edit/view action的配置，还可以配置team -> admin/edit/view action：

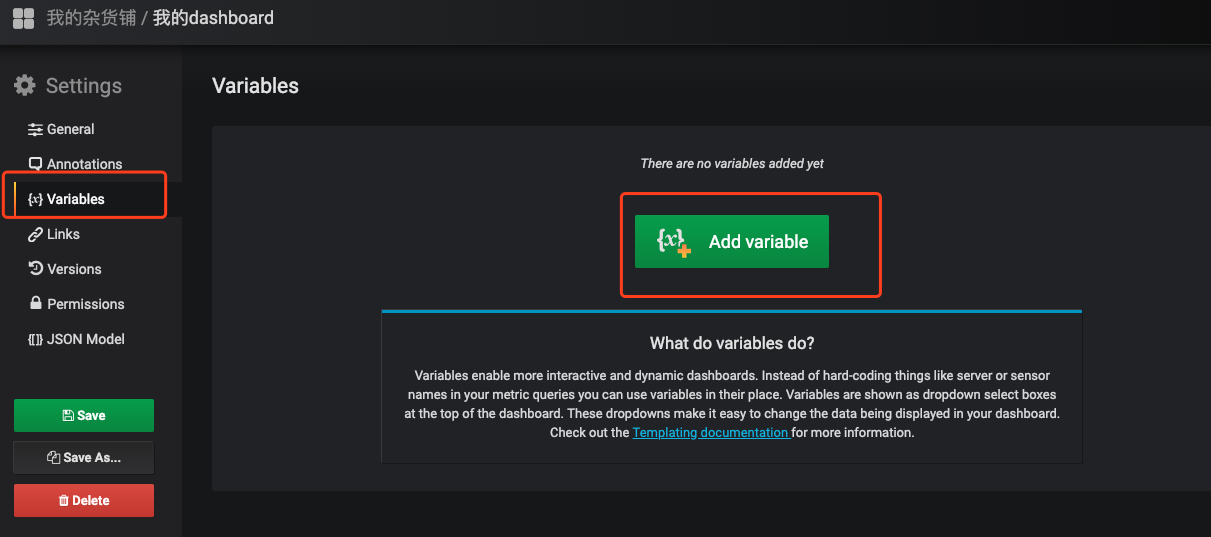
可以给team直接赋予folder或者dashboard的admin/edit/view action权限，这样里面的user就拥有了权限：

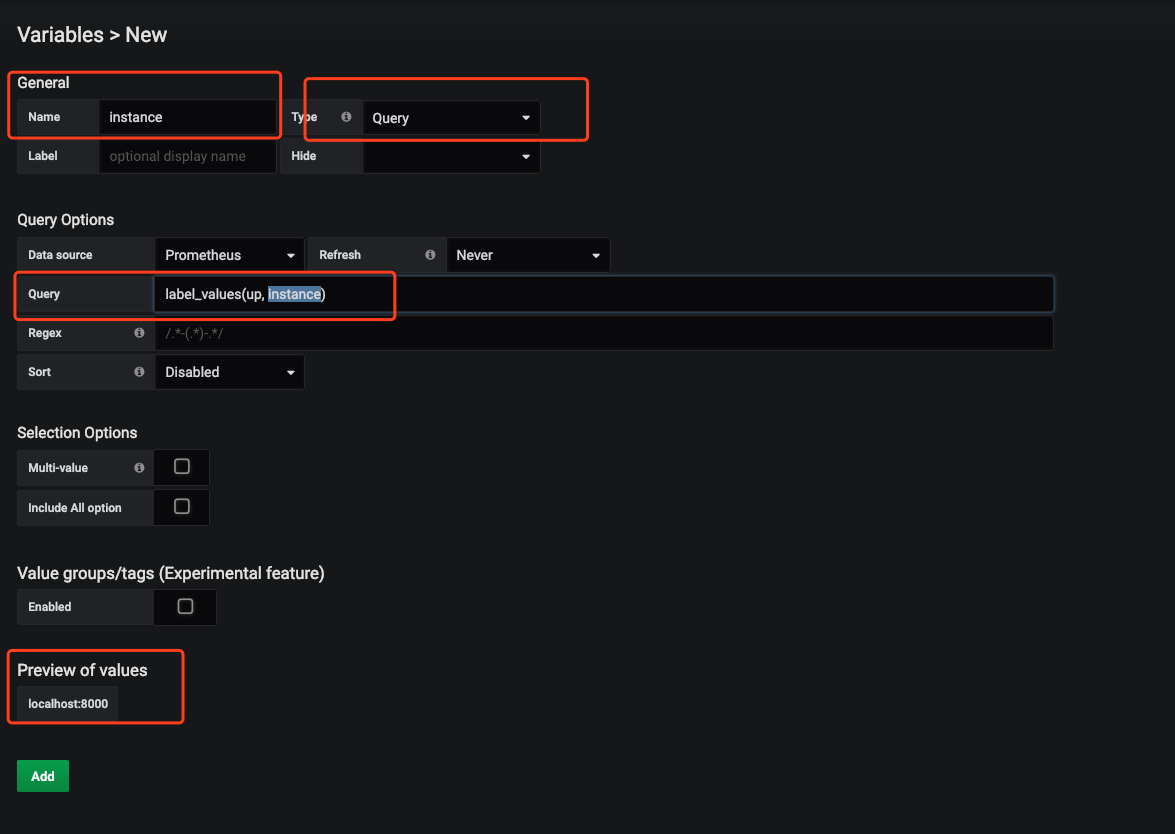


总之，grafana给了你user role以及team两种方式，让你配置对folder或者folder内dashboard的action权限。

## 模板变量

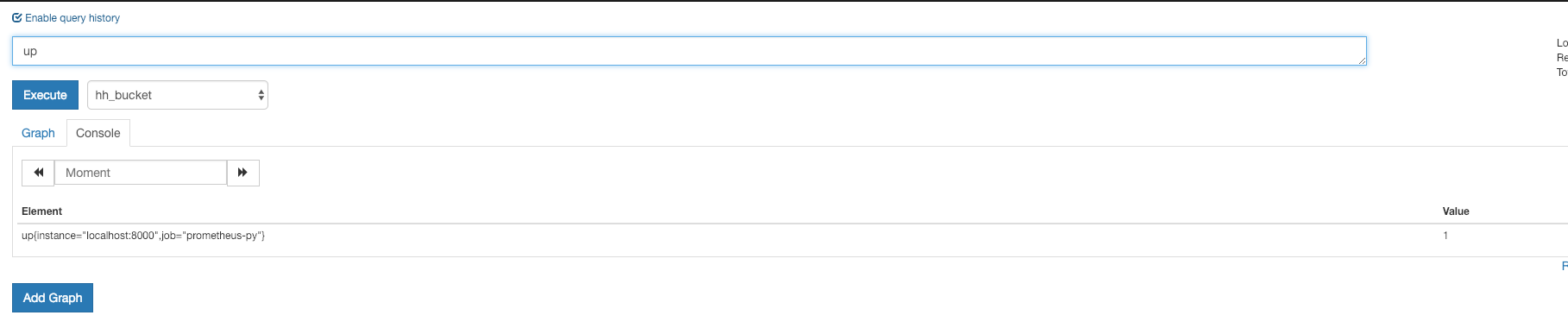
编辑dashboard的variables：



添加一个模板变量：

通过执行promql：”up”，获得相关metrics。

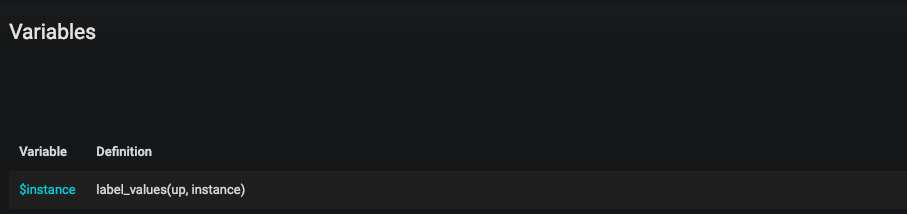
prometheus返回：



利用grafana内置的label\_values方法，可以从metrics中取出instance标签的所有values枚举。

上述最下面的框显示了instance标签的value可选值只有一个：localhost:8000。

现在添加好了：

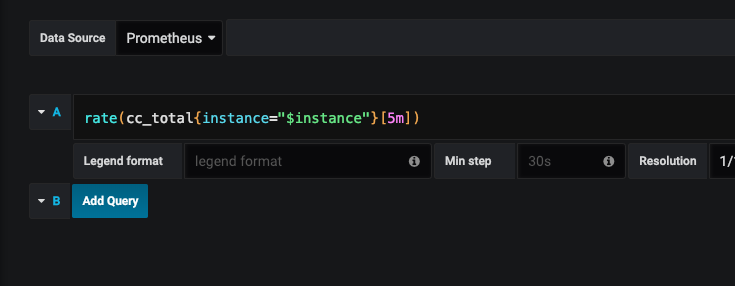


现在dashboard就有选项了：



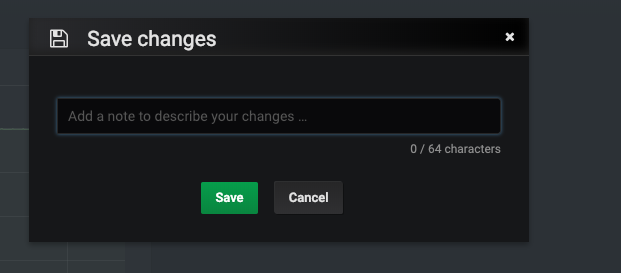
如果我们有多个prometheus client上报，那么instance下拉菜单就会有多个选项。

当选中某个instance的时候，我们想只看它的报表，所以需要改一下panel的promql使用模板变量：



还是promql语法，只是增加了对instance标签的过滤，使用了选中的模板变量。

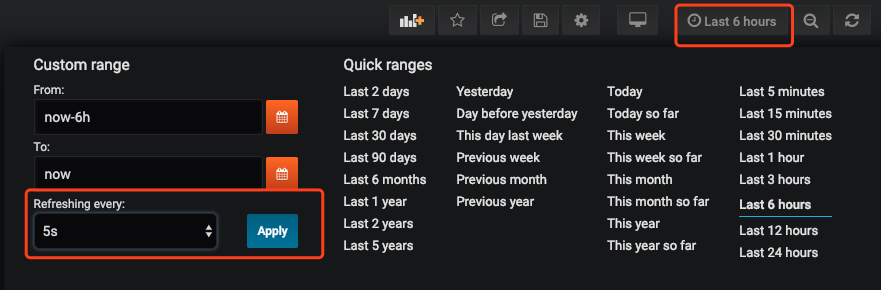
重新保存一下dashboard到数据库，每次可以写一个提交日志：



关于label\_values函数的用法，参考grafana文档： <http://docs.grafana.org/v3.1/datasources/prometheus/>

## 自动刷新

可以让dashboard定时刷新，基于最新时间重新请求prometheus获取最新曲线数据：



# 完