本节所讲内容：

**1.1 Prometheus和Grafana概述**

**1.2 安装配置Prometheus**

**1.3 配置linux node节点和mysql节点**

**1.4 实战：使用Grafana可视化工具美化监控数据显示效果**



## 1.1 Prometheus和Grafana概述

### 1.1.1 Prometheus概述

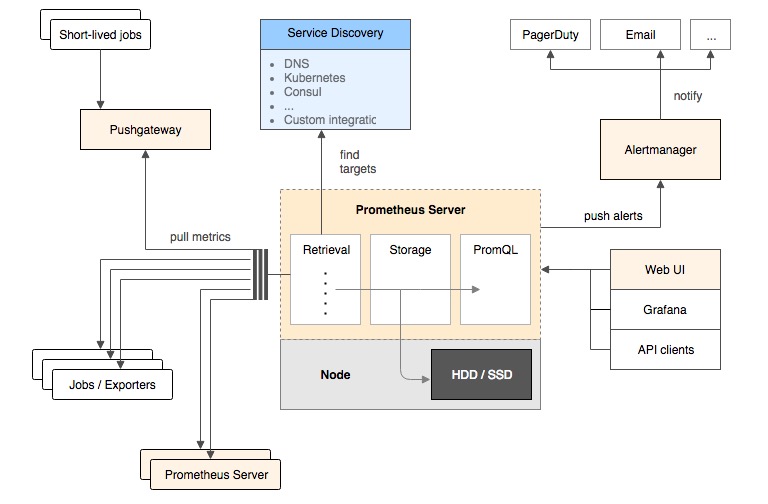
prometheus（普罗米修斯）是一套开源的监控&报警&时间序列数据库的组合，起始是由SoundCloud公司开发的。现在最常见的Docker、Mesos、Kubernetes容器管理系统中，通常会搭配Prometheus进行监控。

Prometheus基本原理是通过HTTP协议周期性抓取被监控组件的状态，这样做的好处是任意组件只要提供HTTP接口就可以接入监控系统，不需要任何SDK或者其他的集成过程。这样做非常适合虚拟化环境比如VM或者Docker 。

输出被监控组件信息的HTTP接口被叫做exporter。目前互联网公司常用的组件大部分都有exporter可以直接使用，比如Varnish、Haproxy、Nginx、MySQL、Linux 系统信息 (包括磁盘、内存、CPU、网络等等)，具体支持的源看：https://github.com/prometheus。

### 1.1.2 Prometheus 架构图解

普罗米修斯（Prometheus）及其一些生态系统组件的整体架构：



Prometheus各组件运行流程如下：

1. Prometheus Server：Prometheus Sever是Prometheus组件中的核心部分，负责实现对监控数据的获取，存储及查询。Prometheus Server可以通过静态配置管理监控目标，也可以配合使用Service Discovery(服务发现)的方式动态管理监控目标，并从这些监控目标中获取数据。其次Prometheus Sever需要对采集到的数据进行存储，Prometheus Server本身就是一个实时数据库，将采集到的监控数据按照时间序列的方式存储在本地磁盘当中。Prometheus Server对外提供了自定义的PromQL，实现对数据的查询以及分析。另外Prometheus Server的联邦集群能力可以使其从其他的Prometheus Server实例中获取数据。

2、Exporters：Exporter将监控数据采集的端点通过HTTP服务的形式暴露给Prometheus Server，Prometheus Server通过访问该Exporter提供的Endpoint端点，即可以获取到需要采集的监控数据。可以将Exporter分为2类：

(1)、直接采集：这一类Exporter直接内置了对Prometheus监控的支持，比如cAdvisor，Kubernetes，Etcd，Gokit等，都直接内置了用于向Prometheus暴露监控数据的端点。

（2）、间接采集：原有监控目标并不直接支持Prometheus，因此需要通过Prometheus提供的Client Library编写该监控目标的监控采集程序。例如：Mysql Exporter，JMX Exporter，Consul Exporter等。

3、AlertManager：在Prometheus Server中支持基于Prom QL创建告警规则，如果满足Prom QL定义的规则，则会产生一条告警。常见的接收方式有：电子邮件，webhook 等。

4、PushGateway:Prometheus数据采集基于Prometheus Server从Exporter pull数据，因此当网络环境不允许Prometheus Server和Exporter进行通信时，可以使用PushGateway来进行中转。

Prometheus的工作流：

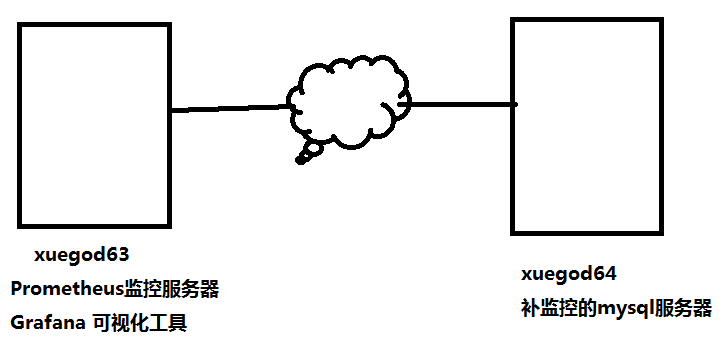
1.Prometheus server定期从配置好的jobs和exporters中拉取metrics，或者接收来自Pushgateway发送过来的metrics，或者从其它的Prometheus server中拉metrics。

2.Prometheus server在本地存储收集到的metrics，并运行定义好的alerts.rules，记录新的时间序列或者向Alert manager推送警报。

3.Alertmanager根据配置文件，对接收到的警报进行处理，发出告警。

4.在图形界面中，可视化采集数据。

实验拓扑图：



## 1.2 安装配置Prometheus

### 1.2.1 安装配置Prometheus

安装go语言环境

[root@xuegod63 ~]#yum install -y epel-release

[root@xuegod63 ~]#yum install go -y

[root@xuegod63 ~]#go version

[root@xuegod63 ~]# tar zxvf prometheus-2.7.1.linux-amd64.tar.gz -C /usr/local/

[root@xuegod63 ~]# cd /usr/local/prometheus-2.7.1.linux-amd64/

修改prometheus配置文件，配置监控linux主机和mysql数据库，配置文件为yml语法，注意缩进对齐

[root@xuegod63 prometheus-2.7.1.linux-amd64]# vim prometheus.yml #在文档的最后插入以下内容

- job\_name: system-status

static\_configs:

- targets: ['192.168.1.64:9100']

labels:

instance: xuegod64

- job\_name: mysql-status

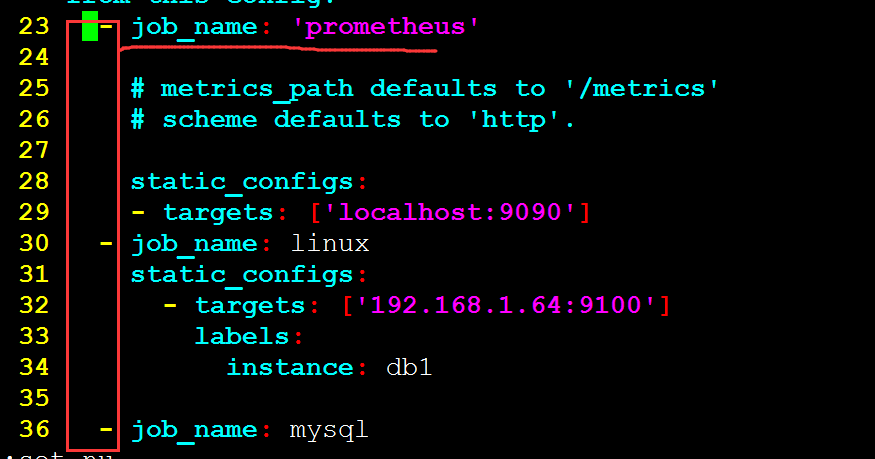
static\_configs:

- targets: ['192.168.1.64:9104']

labels:

instance: xuegod64-mysql

插入时，要和文件之前的job\_name保持对齐，如下图：



注释：

- job\_name: system-status #每一个job\_name都是一个监控项，这里代表监控操作系统状态

static\_configs:

- targets: ['192.168.1.64:9100'] #被监控主机IP地址和端口

labels:

instance: xuegod64 #实例名称，在grafana中表示对应的主机。

- job\_name: mysql-status #监控mysql数据库状态

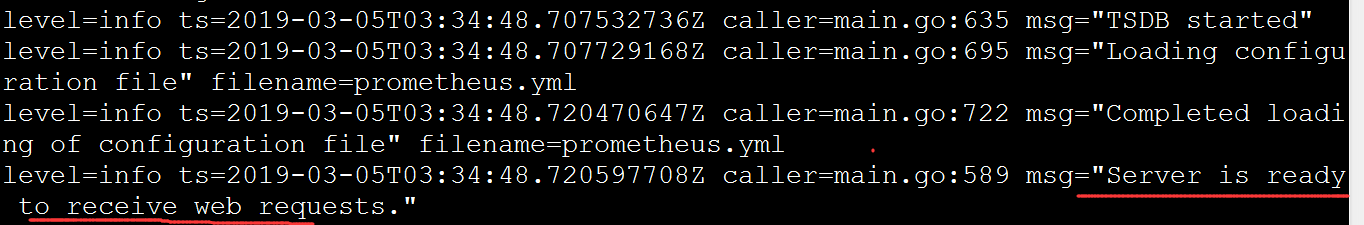
static\_configs:

- targets: ['192.168.1.64:9104'] #目标主机IP地址和端口

labels:

instance: xuegod64-mysql #实例名称，在grafana中表示对应的主机。

[root@xuegod63prometheus-2.7.1.linux-amd64]# ./prometheus --config.file=prometheus.yml



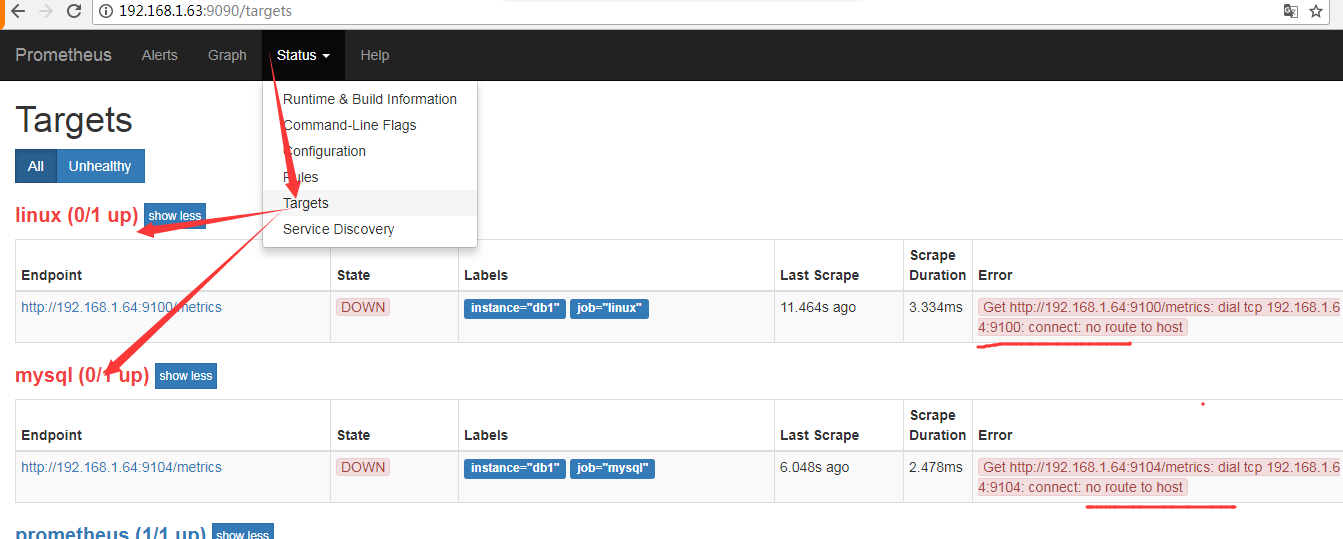
出现以下信息吧表示启动成功。

或直接后台运行：

[root@xuegod63prometheus-2.7.1.linux-amd64]# ./prometheus --config.file=prometheus.yml &

[root@xuegod63 ~]# systemctl stop firewalld && systemctl disable firewalld #关闭防火墙

浏览器输入[http://192.168.1.63:9090](http://192.168.1.64:9090)



注：因为我们还没有开始配置xuegod64了，所以现在还获取不到xuegod64的信息

## 1.3 在xuegod64上配置linux node节点和mysql节点

1、配置linux node节点

[root@xuegod64 ~]# systemctl stop firewalld && systemctl disable firewalld #关闭防火墙

上传node\_exporter-0.17.0.linux-amd64.tar.gz到linux上。

[root@xuegod64 ~]#tar zxvf /root/node\_exporter-0.17.0.linux-amd64.tar.gz -C /usr/local/

[root@xuegod64 ~]# /usr/local/node\_exporter-0.17.0.linux-amd64/node\_exporter &

2、配置mysql node节点

[root@xuegod64 ~]# yum install mariadb-server mariadb -y

[root@xuegod64 ~]# systemctl start mariadb

[root@xuegod64 ~]# mysql #直接登录mysql

mysql> GRANT REPLICATION CLIENT,PROCESS ON \*.\* TO 'mysql\_monitor'@'localhost' identified by '123456';

mysql> GRANT SELECT ON \*.\* TO 'mysql\_monitor'@'localhost';

MariaDB [(none)]> exit;

[root@xuegod64 ~]# tar zxvf /root/mysqld\_exporter-0.11.0.linux-amd64.tar.gz -C /usr/local/

[root@xuegod64 ~]# vim /usr/local/mysqld\_exporter-0.11.0.linux-amd64/.my.cnf #创建隐藏文件.my.cnf，写入以下内容：

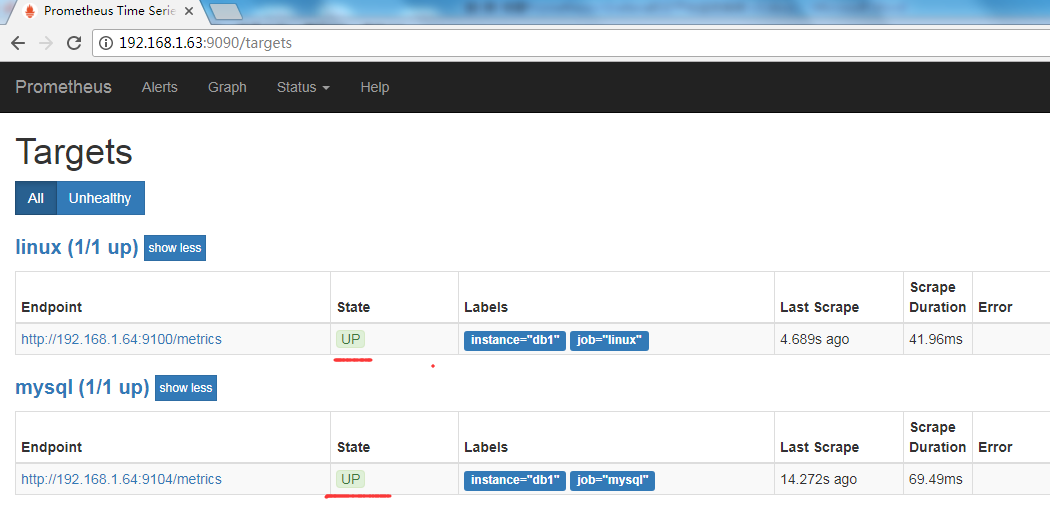
[client]

user=mysql\_monitor

password= [123456](mailto:Nsvicp@c1.com)

[root@xuegod64 ~]#/usr/local/mysqld\_exporter-0.11.0.linux-amd64/mysqld\_exporter --config.my-cnf="/usr/local/mysqld\_exporter-0.11.0.linux-amd64/.my.cnf" &

在web界面，再次查看监控信息已经成功添加linux 和mysql节点，已经up了



## 1.4 配置grafana仪表盘

### 1.4.1 配置grafana

上传grafana到linux上

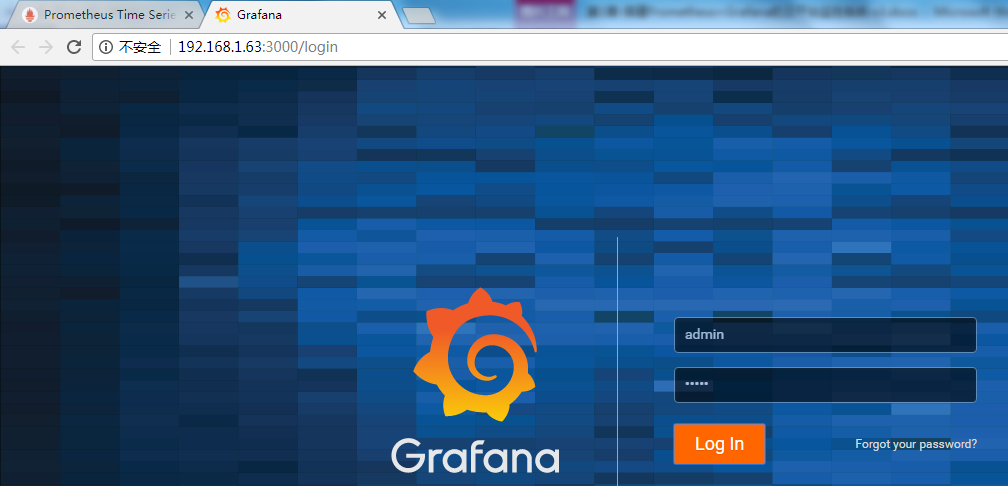
或在线下载：

[root@xuegod63 ~]#wget <https://dl.grafana.com/oss/release/grafana-5.4.3-1.x86_64.rpm>

[root@xuegod63 ~]# yum install -y ./grafana-5.4.3-1.x86\_64.rpm

[root@xuegod63 ~]#systemctl start grafana-server

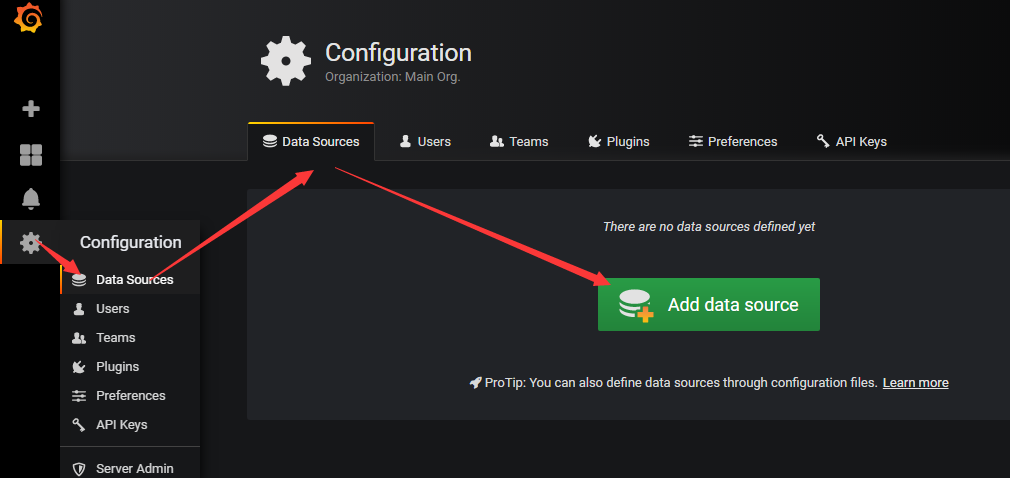
访问：http://192.168.1.63:3000/login 用户名：admin密码admin

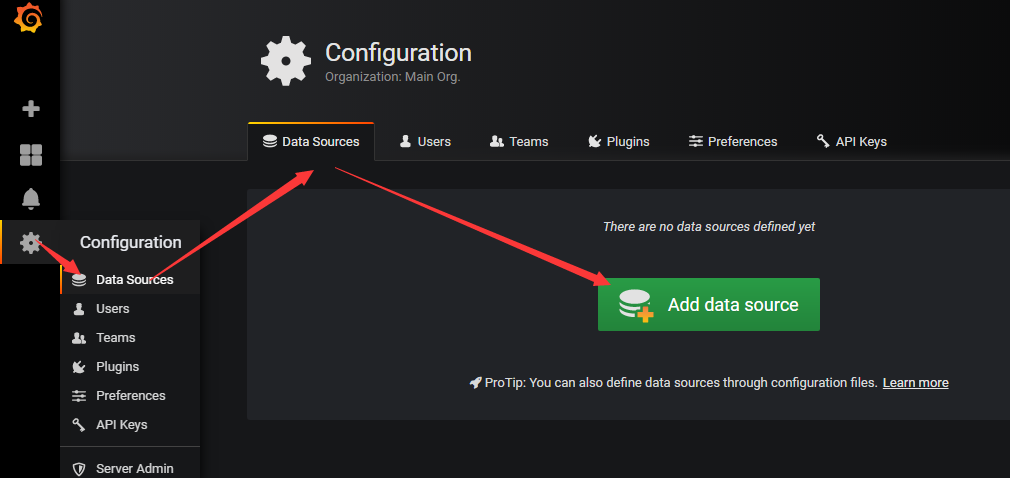


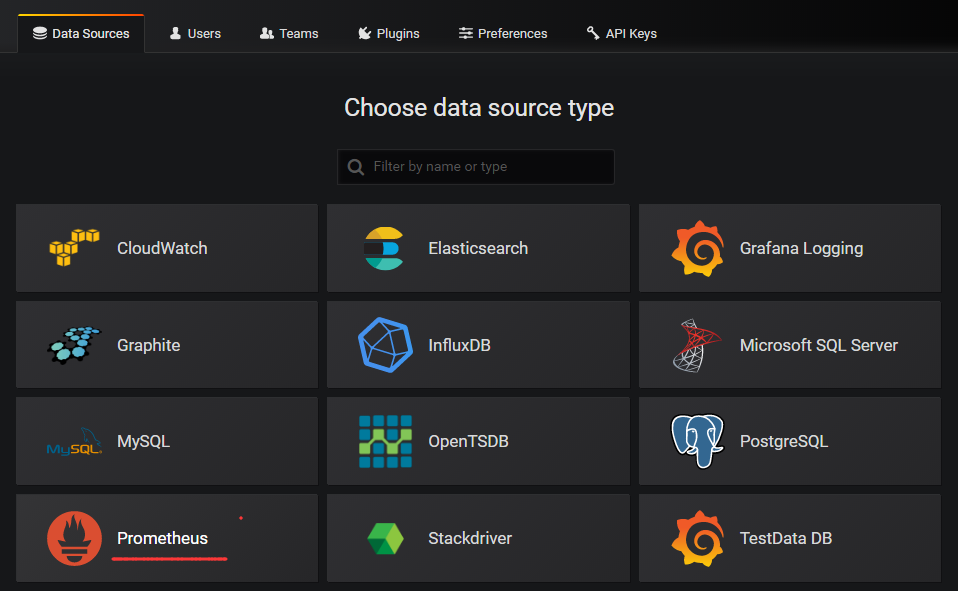
第一次登陆会提示修改密码，咱们改成123456

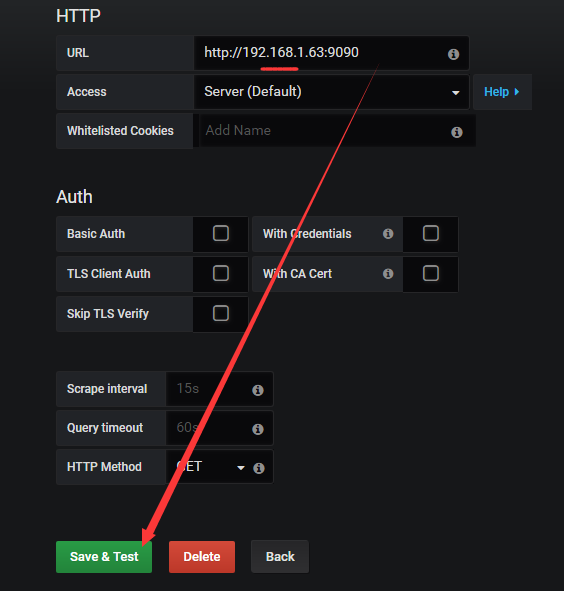


配置数据源

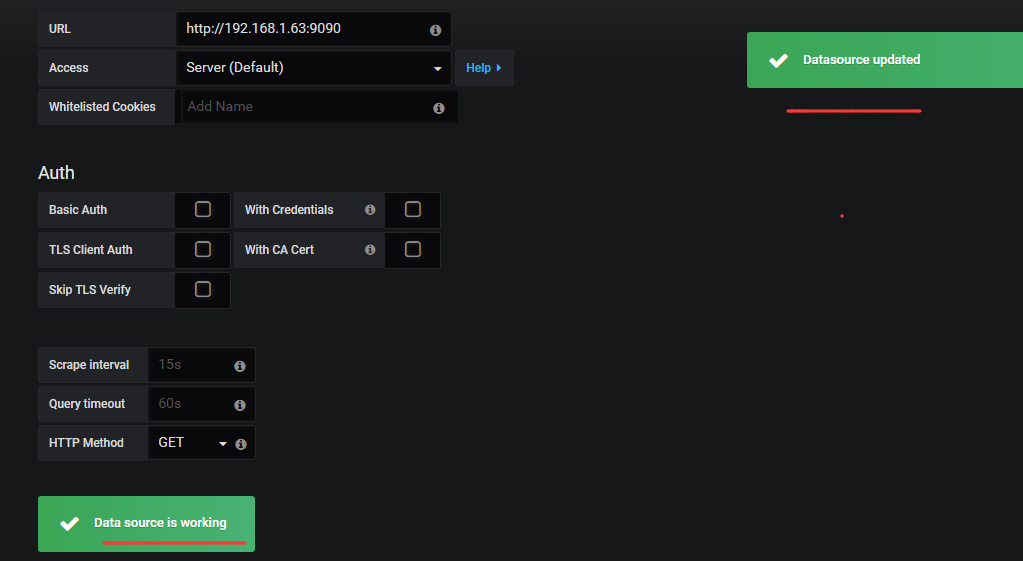






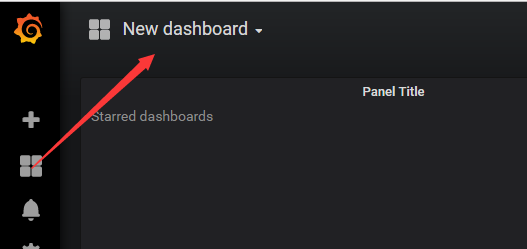


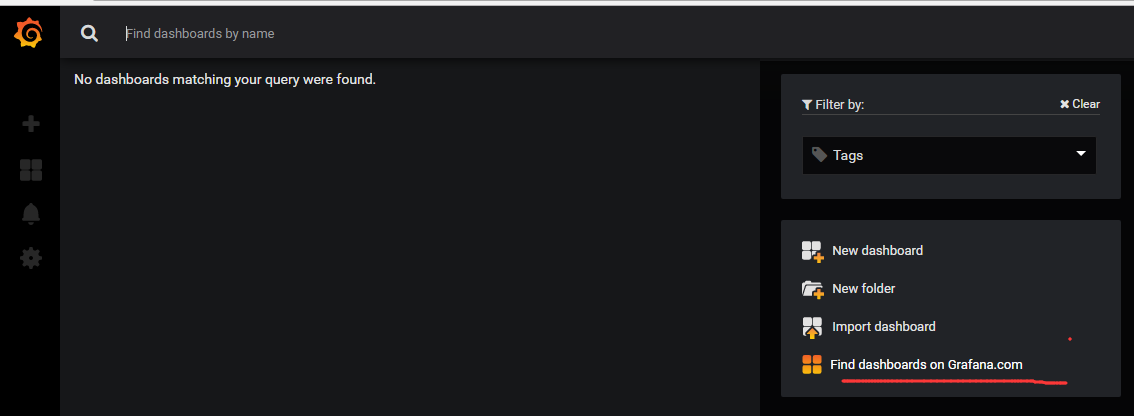
弹出下面信息，说明成功了



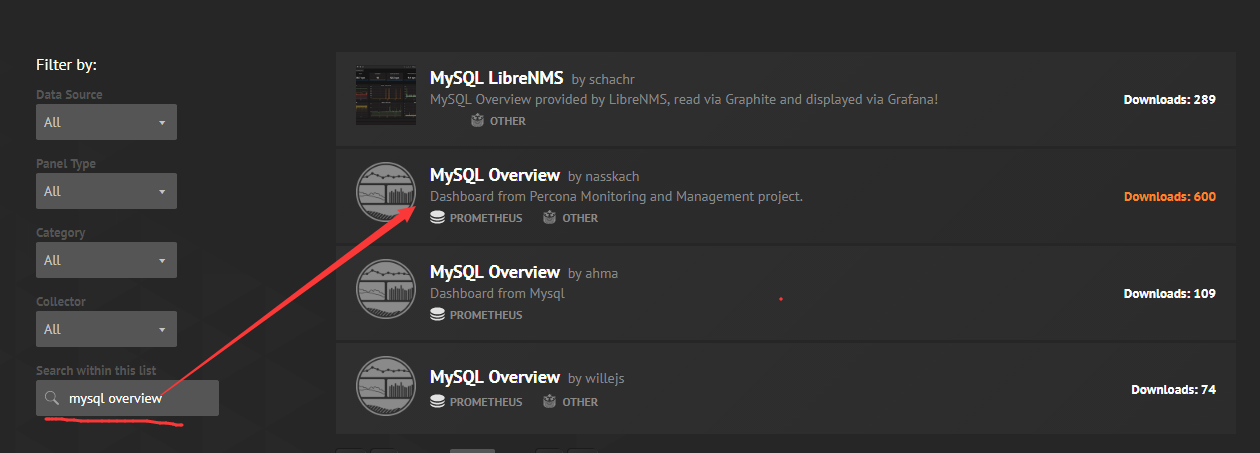
### 1.4.2 把mysql相关的web仪表盘插件导入

先找查出mysql的 web仪表盘插件的ID号

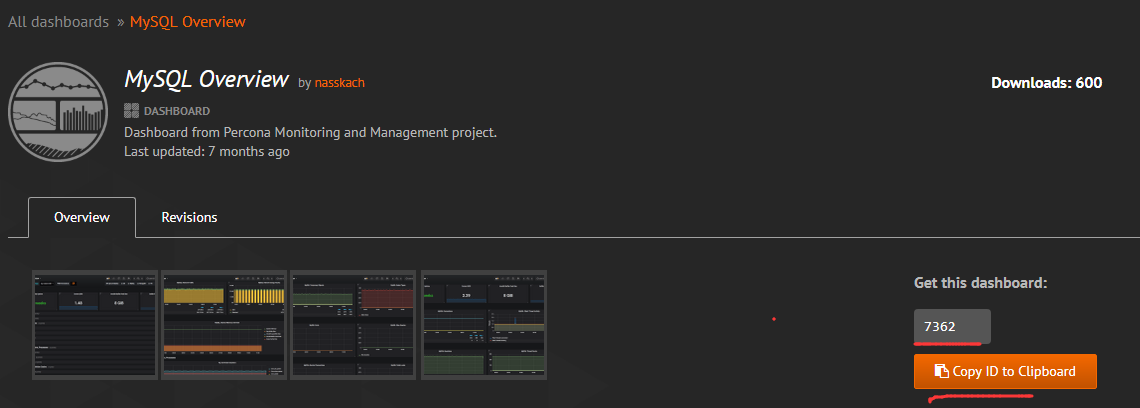




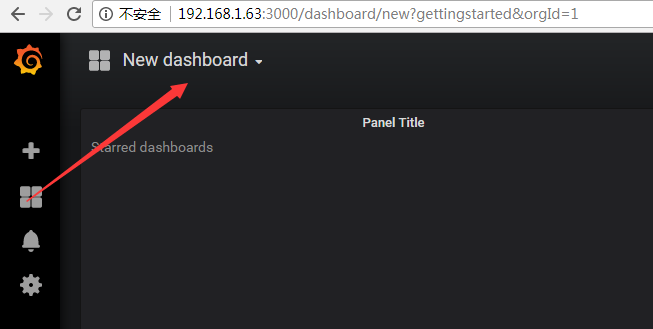
在弹出的web界面，搜索“mysql overview” ，然后选择这个由nasskach开发的界面

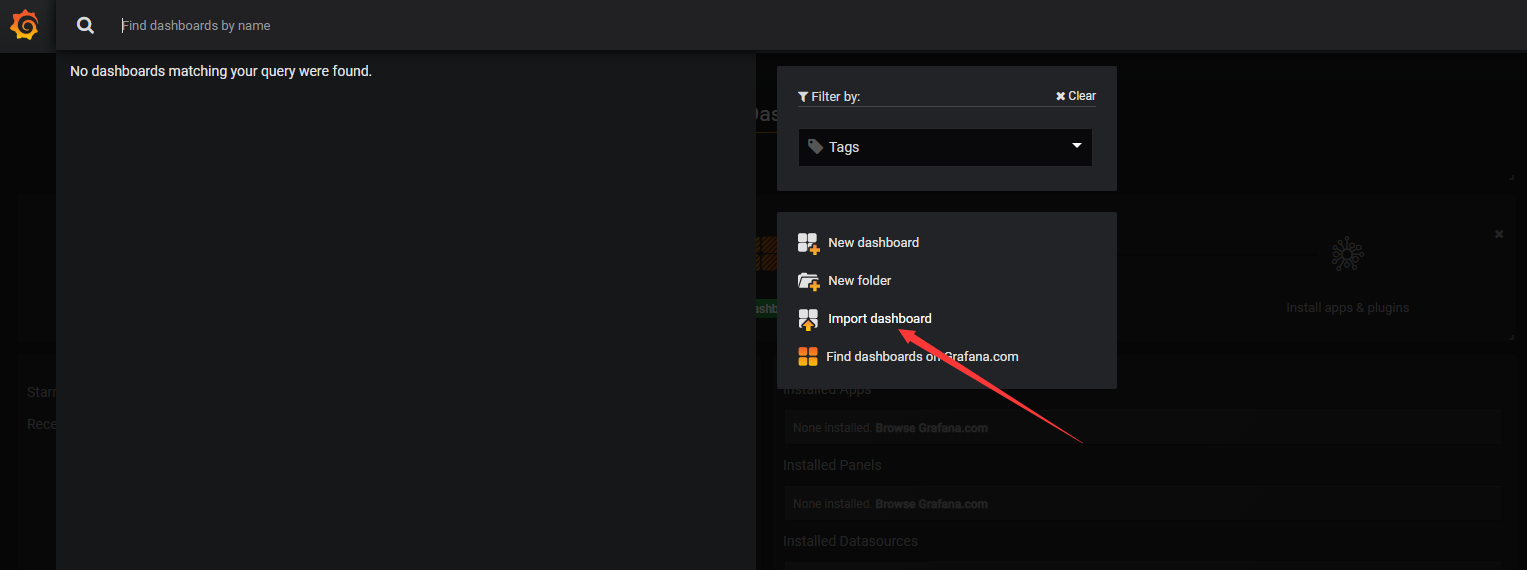


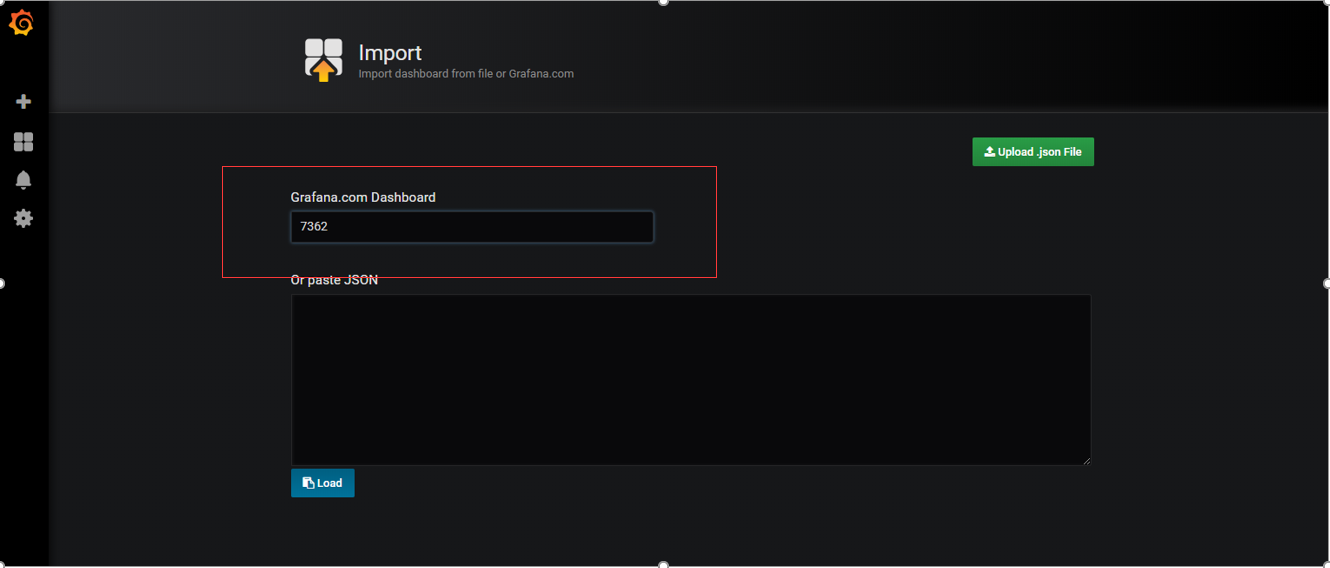
复制7362 的ID



来到grafana主界面，导入dashboard：

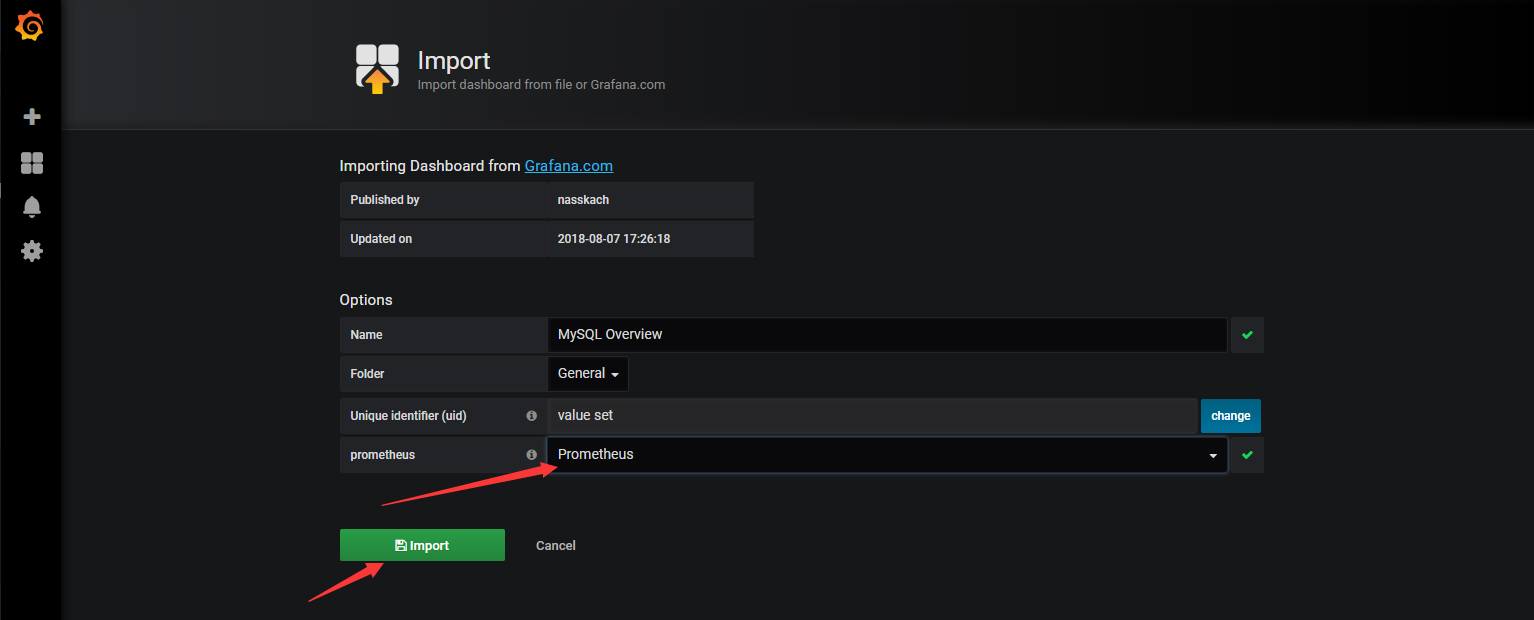


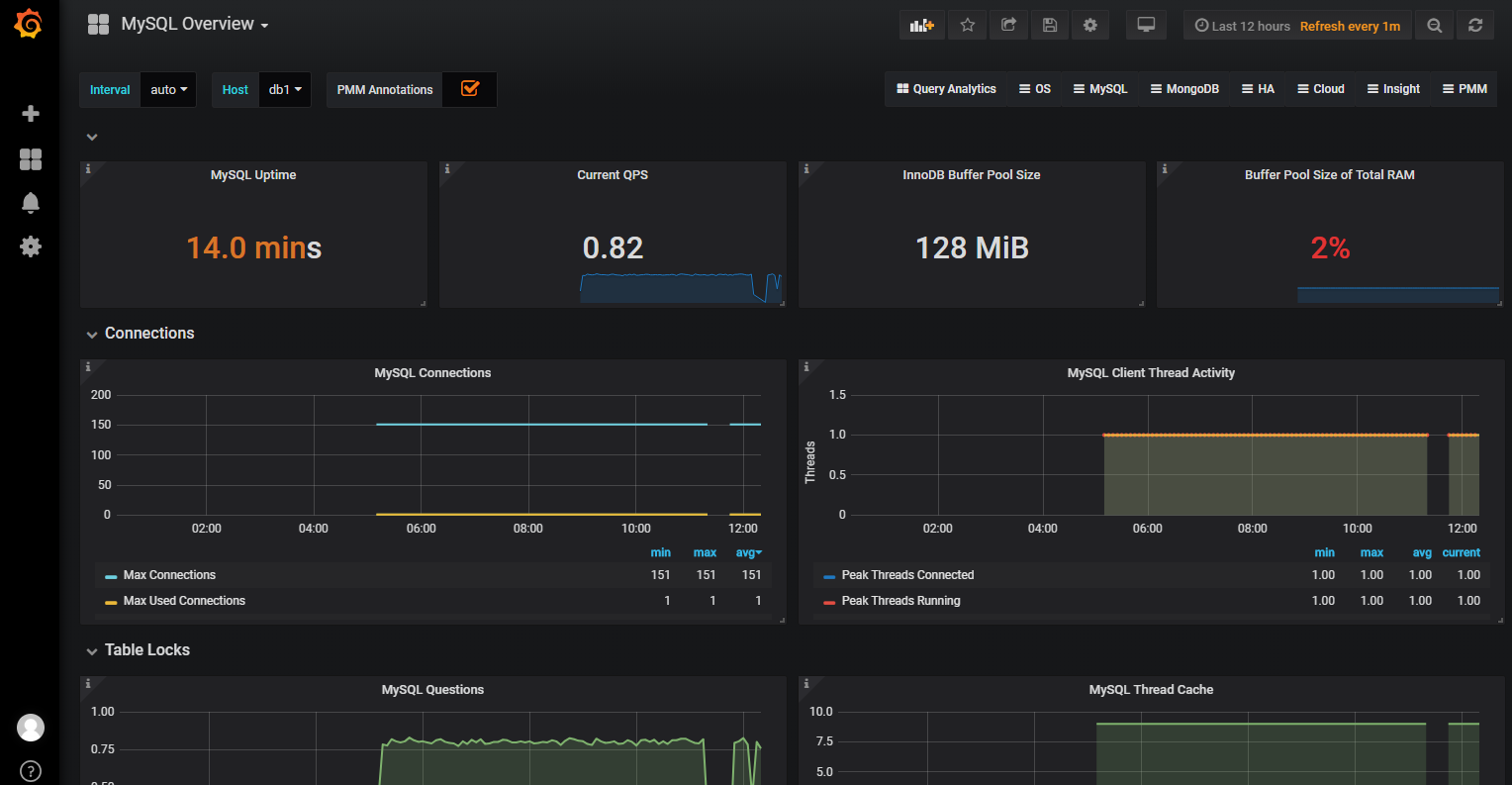




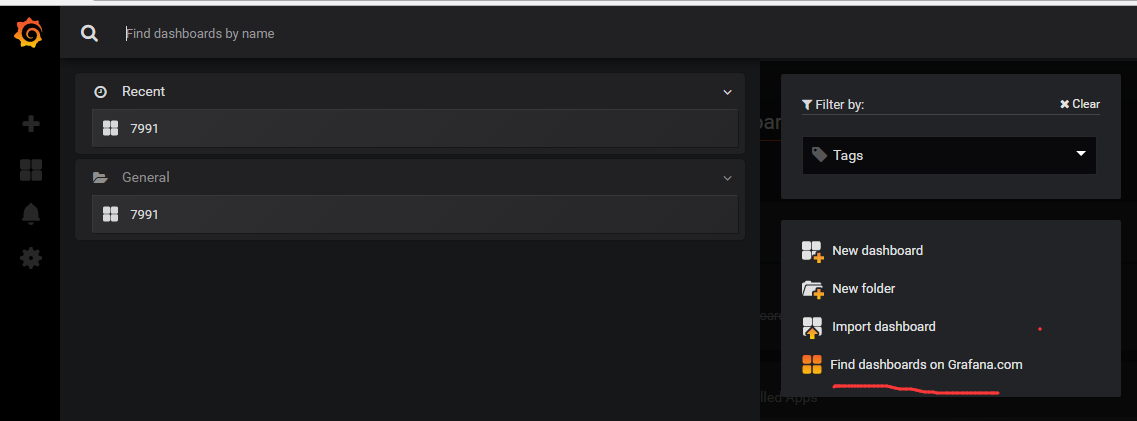
load

下拉选择Prometheus

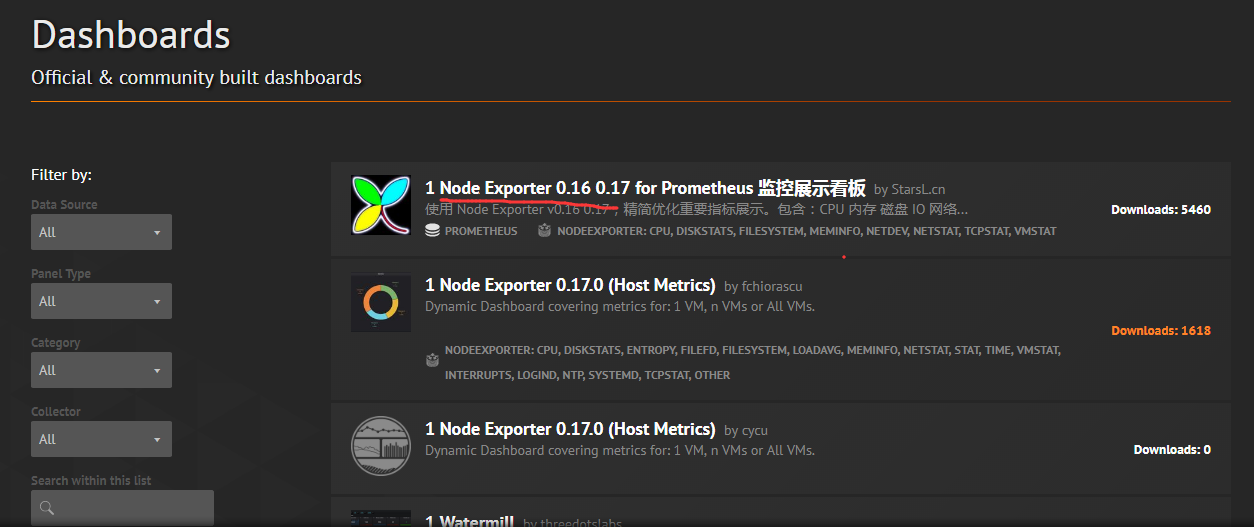




配置linux节点



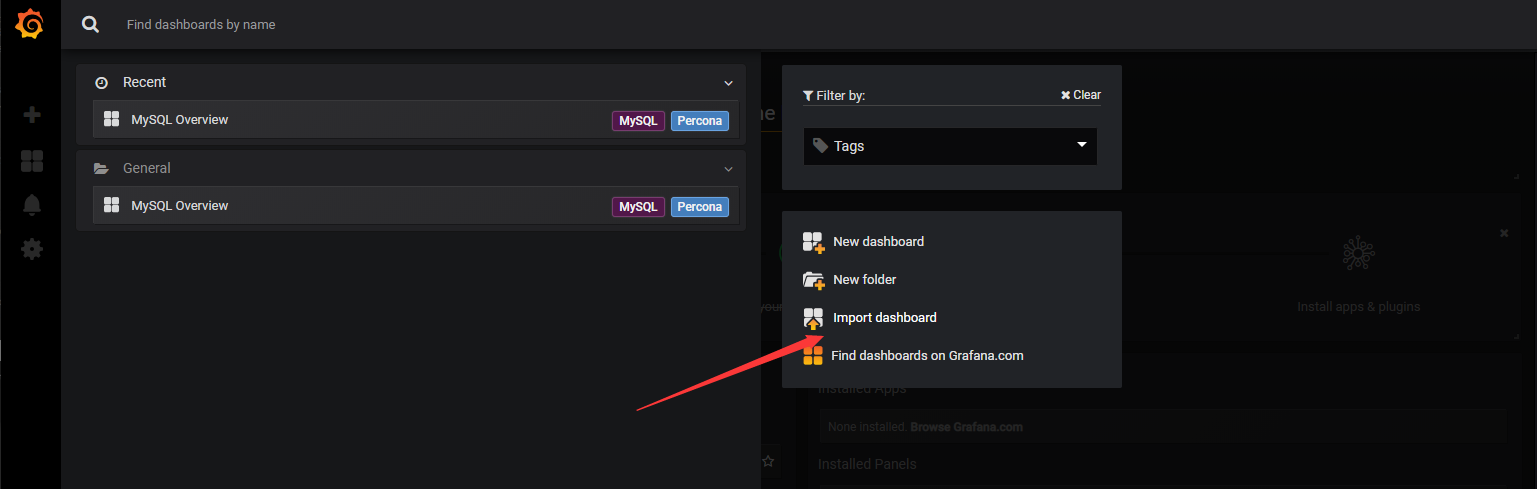
默认的第一个插件就是了：

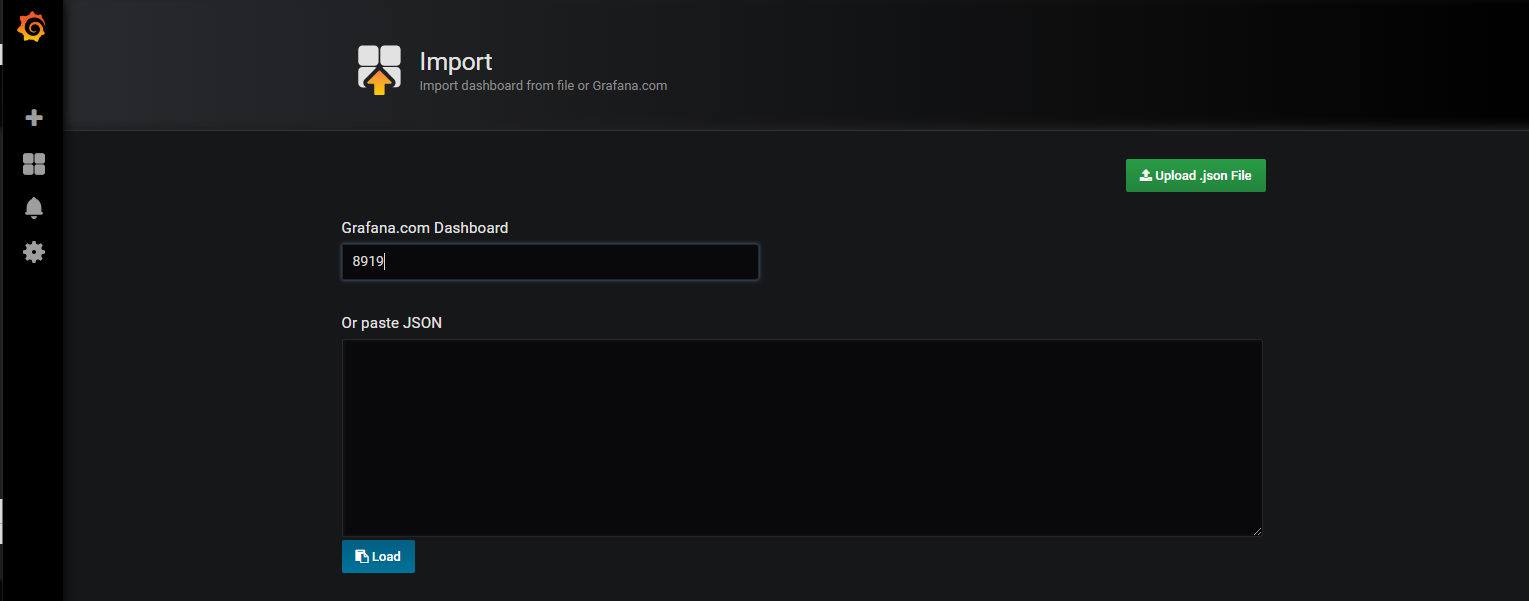




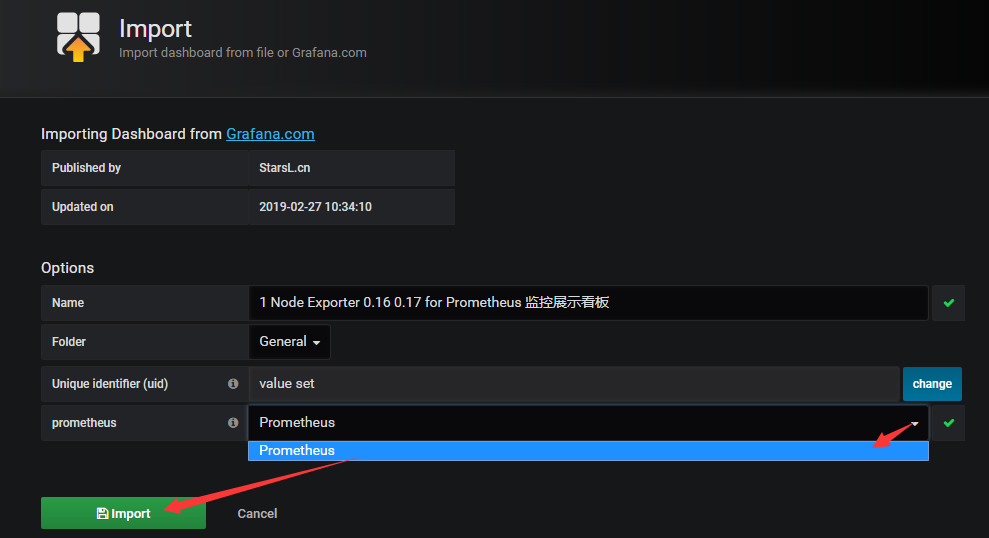
ID号是： 8919

开始导入：





点load，





注：此时该仪表盘并没有图形。

### 1.4.3 安装饼图插件

[root@xuegod63 ~]#grafana-cli plugins install grafana-piechart-panel

[root@xuegod63 ~]#systemctl restart grafana-server.service

刷新浏览器界面：



总结：

1 安装配置Prometheus

2 配置linuxnode节点和mysql节点

3 安装配置grafana

4 配置 Grafana 仪表盘

**QQ :2582668807**