**Influxdb v1版本**

Influxdb（时序数据库），常用的一种使用场景：监控数据统计，利用图形化界面（InfluxDB V1一般配合Grafana）按时间记录一些数据（常用的监控数据、埋点统计数据等），然后制作图表做统计。

**influxdb数据模型**

1.database 代表含义：数据库

2.measurement 代表含义：数据库中的表

3.point 代表含义：表里面的一行数据，由时间戳（time）、数据（field）、标签（tags）组成

time 代表含义：每个数据记录时间，是数据库中的主索引（会自动生成）

fields 代表含义：各种记录值（没有索引的属性）（必须要有）

tags 代表含义： 各种有索引的属性（可以没有）

1. series 代表含义：所有在数据库中的数据，都需要通过图表来展示，表示表里面的数据，可以在图表上画成几条线，通过tags排列组合计算出来

**influxdb基本操作**

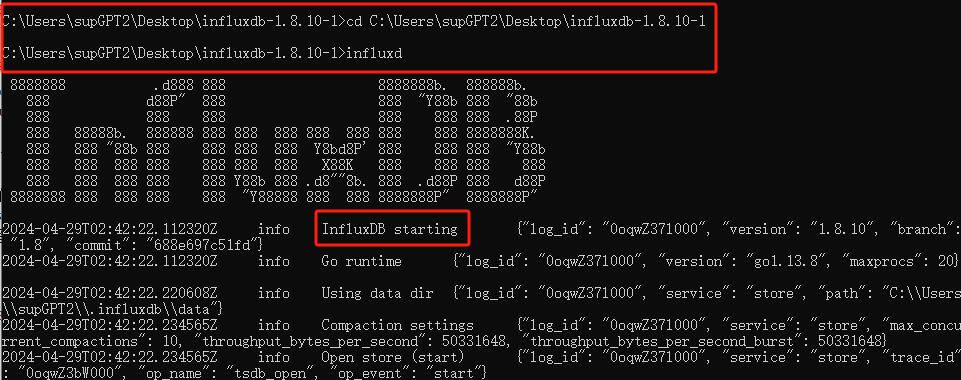
1.下载链接（v1.8版本）

：<https://dl.influxdata.com/influxdb/releases/influxdb-1.8.10_windows_amd64.zip>

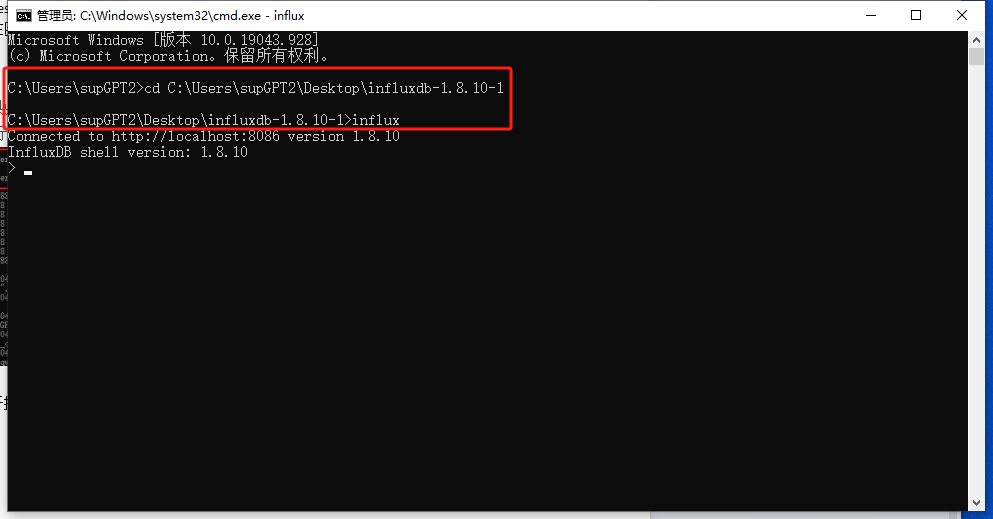
其他版本：

https://github.com/influxdata/influxdb/releases?page=1

2.运行influxdb



打开操作面板：



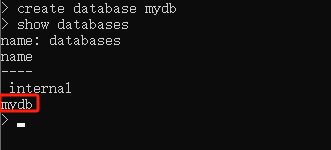
3.数据库操作

创建数据库：

>create database mybd

显示数据库：

>show databases



1. 使用数据库，插入和展示measuremet

插入数据：

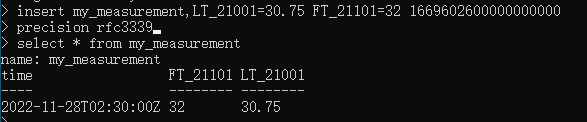
> insert my\_measurement,LT\_21001=30.75 FT\_21101=32 1669602600000000000

my\_measurement：表示插入的表名

示例

LT\_21001=30.75 FT\_21101=32：表示filed key和filed value

1669602600000000000：表示想要记录的时间，使用Unix时间戳作为时间格式



5.批量数据导入

当需要导入大量的数据到数据库mydb中，使用使用代码操作如下：

import csv

from datetime import datetime

from influxdb import InfluxDBClient

import threading

# 连接到InfluxDB数据库

host = 'localhost'

port = 8086

user = 'admin'

password = 'admin'

dbname = 'mydb'

client = InfluxDBClient(host, port, user, password, dbname)

# 读取CSV文件并构建数据点列表

csv\_file = ''

data\_points = []

def process\_column(column\_name):

with open(csv\_file, 'r') as file:

reader = csv.DictReader(file)

for row in reader:

timestamp = datetime.strptime(row['date'], '%Y/%m/%d %H:%M').strftime('%Y-%m-%dT%H:%M:%SZ')

value = float(row[column\_name])

# 构建InfluxDB数据点

data\_point = {

"measurement": "you\_measurement",

"time": timestamp,

"fields": {column\_name: value}

}

# 写入数据到InfluxDB

client.write\_points([data\_point])

# 获取CSV文件的列名

with open(csv\_file, 'r') as file:

reader = csv.DictReader(file)

columns = reader.fieldnames

columns.remove('date') # Remove 'date' column from processing

# 创建并启动线程

threads = []

for column\_name in columns:

thread = threading.Thread(target=process\_column, args=(column\_name,))

threads.append(thread)

thread.start()

# 等待所有线程完成

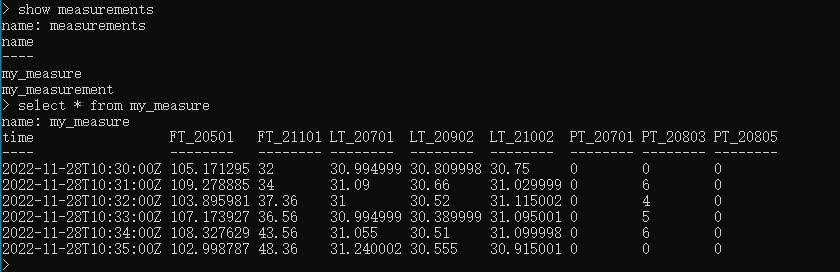
for thread in threads:

thread.join()

# 关闭InfluxDB连接

client.close()

成功将示例文件数据导入数据库mydb中

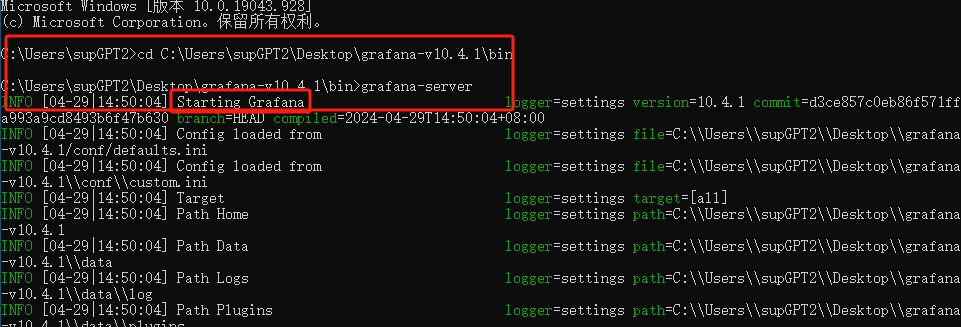


**Grafana**

Grafana 是一个监控仪表系统，它是由 Grafana Labs 公司开源的的一个系统监测 (System Monitoring) 工具。它可以大大简化监控的复杂度，只需要提供所需要监控的数据，就可以生成各种可视化仪表。同时它还有报警功能，可以在系统出现问题时发出通知。

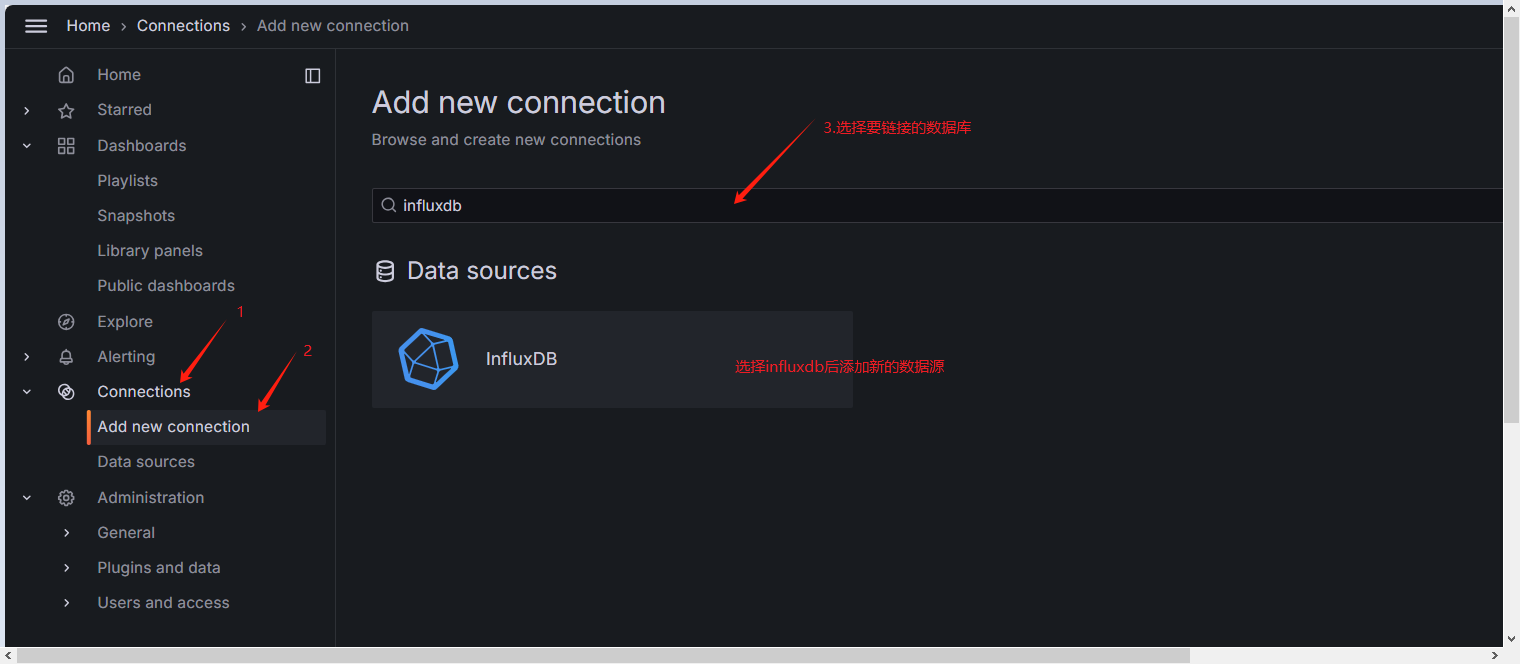
下载地址：<https://grafana.com/grafana/download/10.4.2?platform=windows>

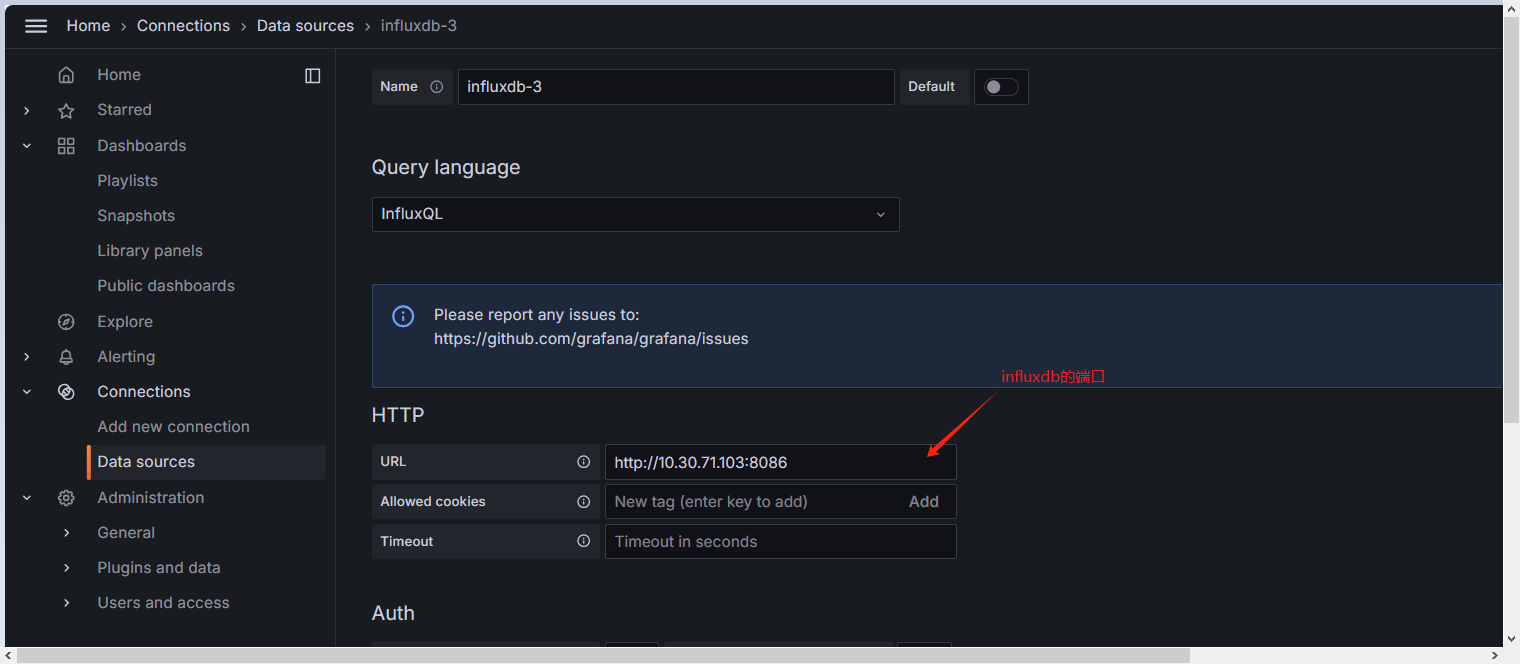
启动grafana

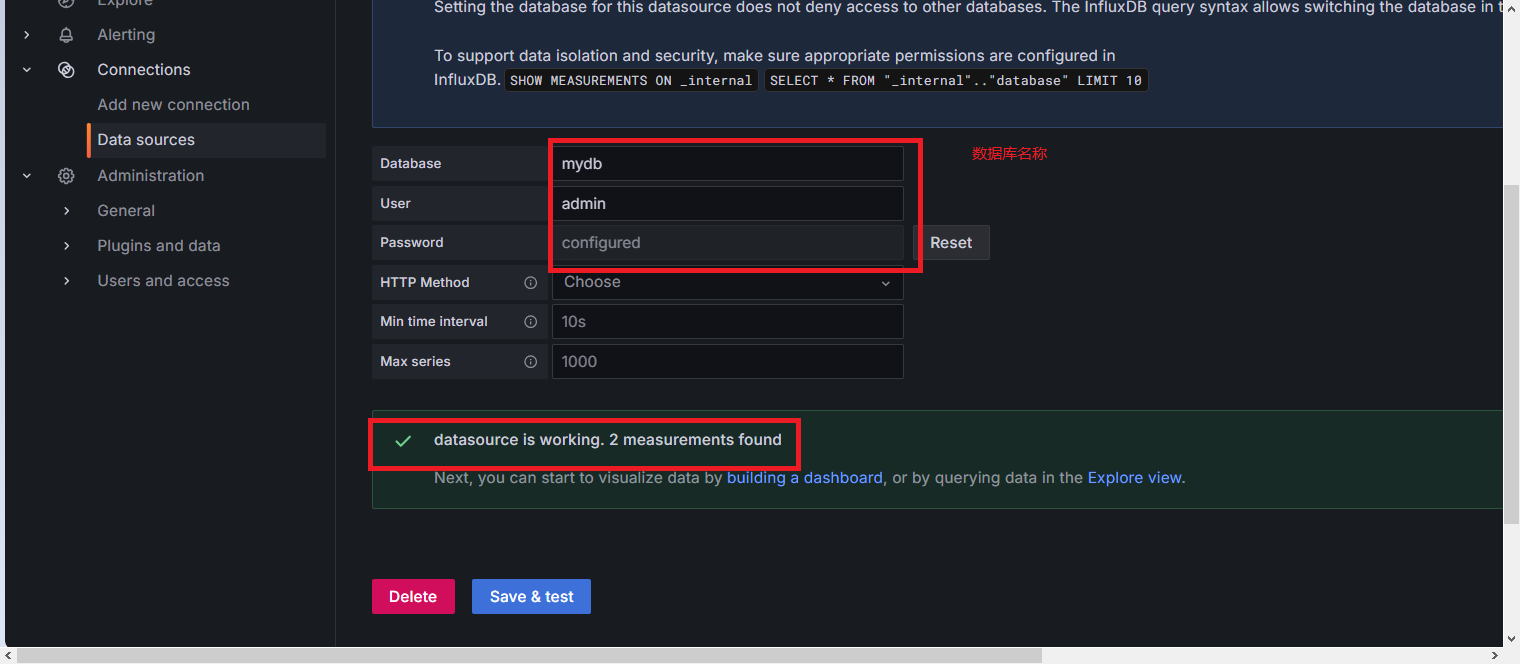


访问grafana：  
http://ip地址:3000  
grafana的默认用户名密码都是admin，第一次登录会要求更改密码

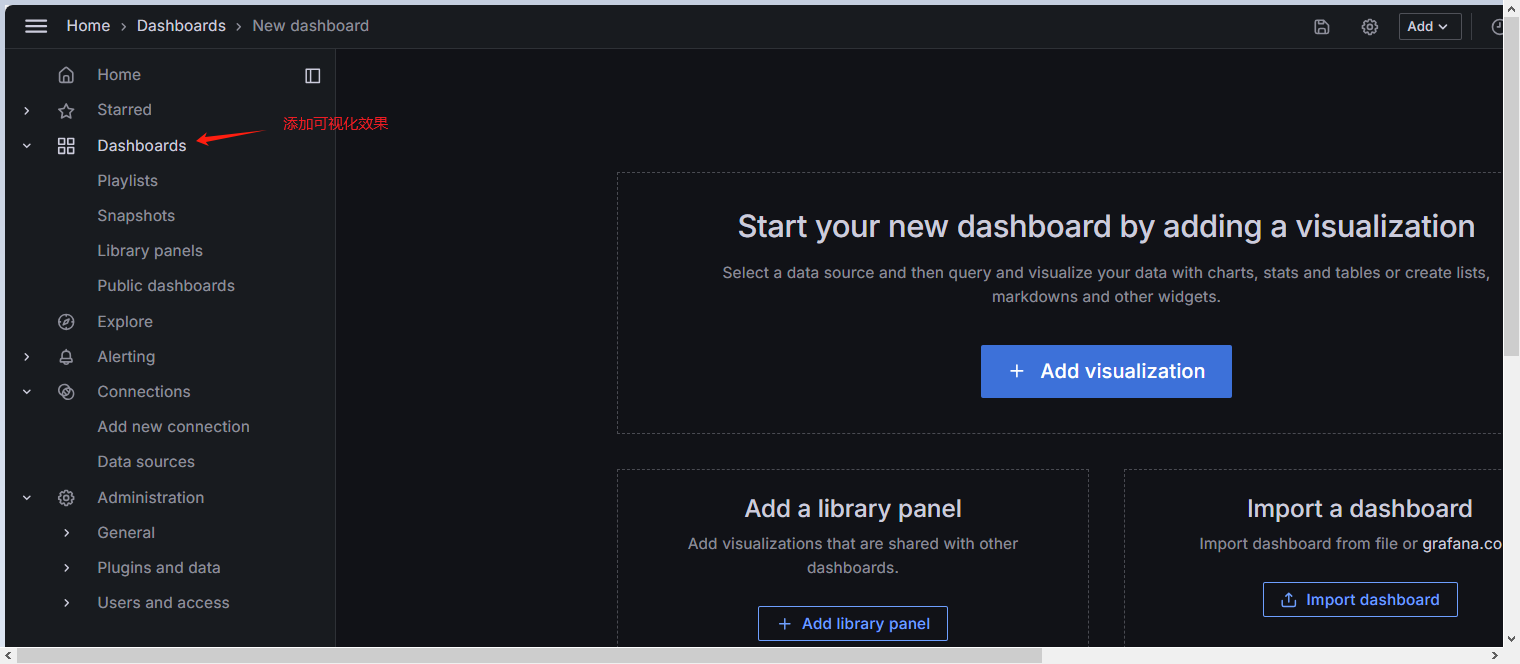
配置influxdb的数据源，用户名和密码

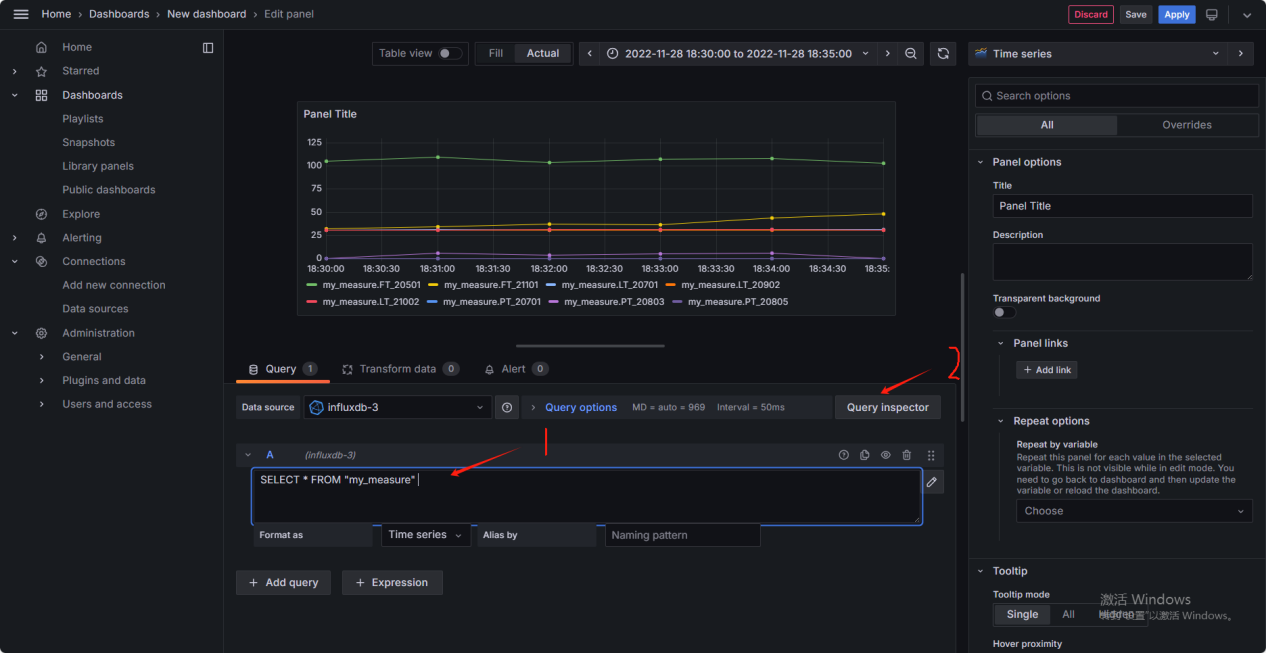






创建仪表板



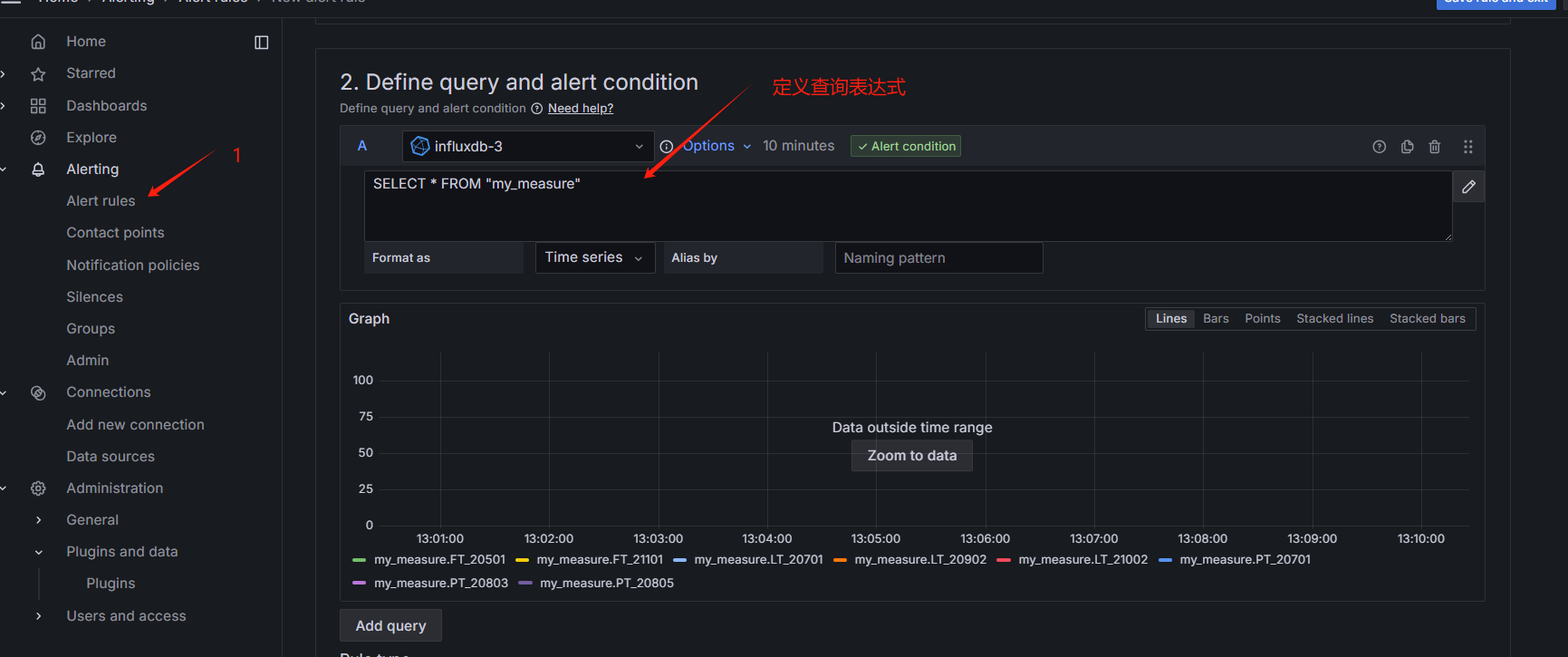


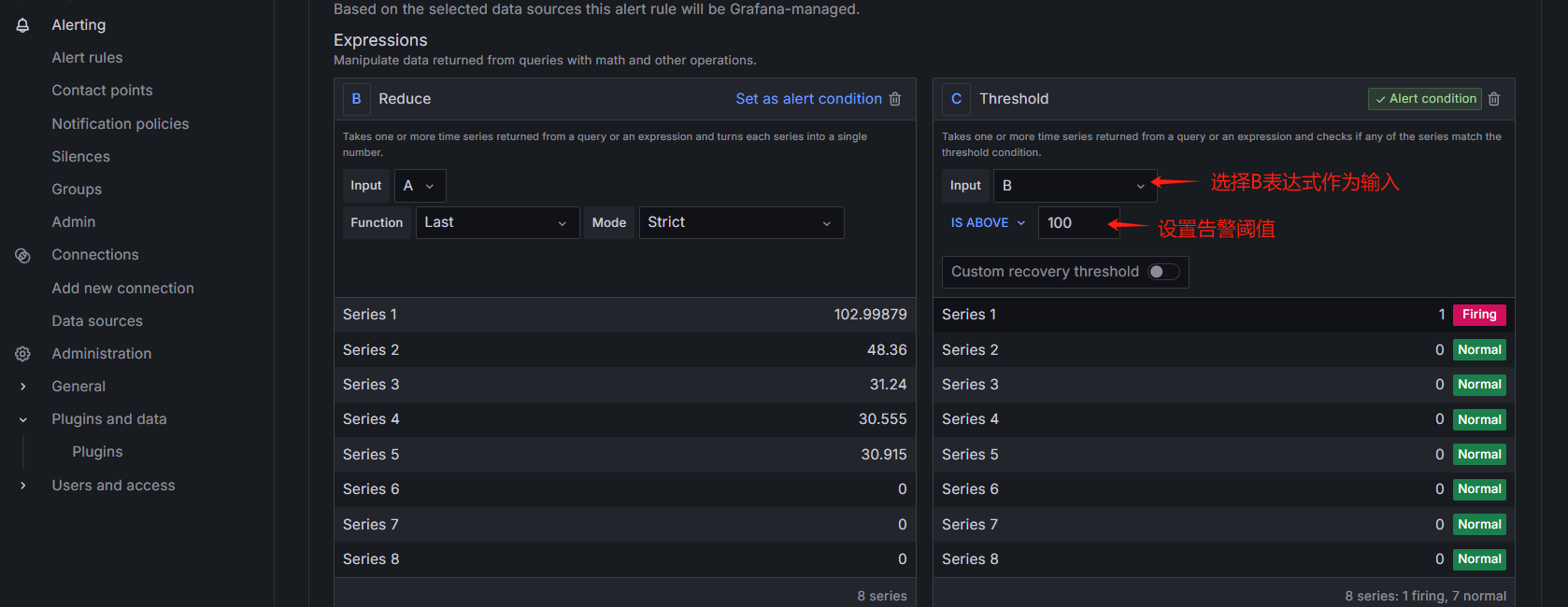
面板展示



告警规则

告警规则是一组确定何时应触发告警的条件。它由一个或多个查询和表达式、需要满足的条件、确定评估告警规则的频率以及必须满足条件才能触发告警的持续时间组成





联络点

通过Grafana告警，配置文件中配置 stmp 服务，配置文件中添加SMTP/Emailing 和 Alerting 模块

[root@server ~]# vim /etc/grafana/grafana.ini

#################################### SMTP / Emailing ########################## [smtp]

enabled = true

host = smtp.qq.com:465 #使用的邮箱

user = xxx@foxmail/qq/163.com

password = \*\*\*\*\*\*\*\*\*\*\* #使用mail授权码

skip\_verify = true

from\_address = xxx@foxmail/qq/163.com

#################################### Alerting ############################ [alerting] enabled = true

execute\_alerts = true

经过上述设置完后重新启动grafana

