# Grafana+Telegraf+infludb编译安装（Centos）

## 一、安装golang 环境

1. 安装go 环境

官网下载：https://dl.google.com/go/go1.12.2.linux-amd64.tar.gz

1. tar zxvf go1.10.1.linux-amd64.tar.gz -C /usr/local
2. 创建项目路径

mkdir -p /usr/local/gohome 创建项目路径

1. 修改环境变量

vim /etc/profile 编辑环境变量，在末尾添加

export PATH=$PATH:/usr/local/go/bin

export GOROOT=/usr/local/go #设置为go安装的路径

export GOPATH=/usr/local/gohome #默认安装包的路径

export PATH=$PATH:$GOROOT/bin:$GOPATH/bin

1. 让环境变量生效，输入命令 source /etc/profile
2. 验证： go version

## 二、安装 node环境

1、下载

wget <https://nodejs.org/dist/v8.5.0/node-v8.5.0-linux-x64.tar.gz>

2、tar zxvf node-v8.5.0-linux-x64.tar.gz

3、配置环境变量

vim /etc/profile 编辑环境变量，在末尾添加

export NODE\_HOME="/usr/local/node-v8.5.0-linux-x64"

export PATH=$PATH:$NODE\_HOME/bin

4、刷新 source /etc/profile

5、验证

node -v

npm –v

## 三、安装git

yum install git

git version

四、编译源码包

自己建目录结构

方法一：

mkdir github.com/grafana/grafana

git clone gitlab.jec.com/jec/dev/code/mygrafana

(如遇509错误 export GIT\_SSL\_NO\_VERIFY=1 环境变量 /etc/profile 最后增加)

go install github.com/grafana/grafana

(can't load package: package mygrafana: build constraints exclude all Go files in /usr/local/gohome/src/mygrafana、)—不用管

方法二：

go get gitlab.com/wolfwolfgod/mygrafana

(can't load package: package mygrafana: build constraints exclude all Go files in /usr/local/gohome/src/mygrafana、)—不用管

(注释：go build : 编译出可执行文件  
go install : go build + 把编译后的可执行文件放到GOPATH/bin目录下  
go get : git clone + go install)

## 四、安装gcc g++

yum install gcc

yum install gcc-c++

五、

cd $GOPATH/src/github.com/grafana/grafana

go run build.go setup

go run build.go build

npm install -g yarn

yarn install --pure-lockfile

yarn start

## 六、安装FPM打包工具

yum -y install ruby rubygems ruby-devel

yum install rpm-build

gem install fpm

打包： go run build.go package

在 grafana/dist 文件中产生 安装包

## 七、rpm 包安装

rpm –ivh \*.rpm包

## 八、安装infludb

### 1、安装

$  wget https://dl.influxdata.com/influxdb/releases/influxdb-1.2.2.x86\_64.rpm  (下载包)

$ yum localinstall influxdb-1.2.2.x86\_64.rpm  (用yum进行本地安装)

$ vim /etc/influxdb/influxdb.conf (修改配置文件)

------------------------------------------------------------------------------------------

reporting-disabled = true ( 这个要设置真，关闭定时上传数据到influxdata.com)

#bind-address = ":8086"(这个需要自己手动添加，指定http的连接操作端口，默认为8086)

[admin]

  # Determines whether the admin service is enabled.

  enabled = true （web管理界面，1.1版本以上默认关闭。需要的话，可以手动打开）

  # The default bind address used by the admin service.

  bind-address = ":8083"    （web服务界面的端口）

### 2，启动influxdb，并使用

## 本人并没有开启web界面，所以这里讲一下命令行的操作。就类似与常用数据库。

$ systemctl start influxdb （启动influxdb）

$ systemctl enable influxdb （设为开机启动）

$ influx ( 进入influxdb )

Connected to http://localhost:8086 version 1.2.2

InfluxDB shell version: 1.2.2

也可以浏览器访问：

http://localhost:8086

-----------------------------------------------influxdb 的基本操作 -----------------------------

>create database test\_db (创建数据库test\_db )

>show databases （列出所有数据库）

>drop database test\_db （删除数据库）

>use test\_db ( 转入test\_db数据库下 )

>create measurement test\_table ( 创建test\_table表，measurements==table(关系型数据库中的) )

>show measurements  （列出所有表）

>drop measyrement test\_table （删除表test\_able）

>insert test\_table,host=web1 cpu=22,memory=33 （插入数据，test\_table为表名，host为除了tag是，也就是方便用于度量的分类标签，可以是字符串。cpu和memory是数值，不能是字符串，只能是数值，中间用空格隔开）格式如下：

insert table\_name,tags=\* values=values,values=values （分别为三段

>curl -i -X POST "http://127.0.0.1:8086/write?db=testDb【&u=username&p=password】" --data-binary "test\_table,host=127.0.0.1 cpu=22,memory=33"

！！！！！！！！！！！！！！！！！！！！！利用http接口远程插入数据，最常用的功能,用密码时，通过u和P的参数进行认证，没有则去掉。也就是利用post提交，默认的连接操作的端口为8086.

> select \* from test\_table （查询语句，类似sql语句）

>select \* from test\_table where cpu = 22 ( where用法，类似sql)

>select \* from test\_table where host = 'web1' （同上）

>select \* from test\_table order by time [desc] （按时间排序，默认为顺序，加上desc为倒序）

>show users （查看用户）

>create user "username" with password 'password' （创建普通用户）

>create user "username" with password 'password' with all privileges (创建管理员用户)

>drop user "username" （删除用户）

>auth （用户认证，用于设置了密码后，登陆influxdb的认证）

username: username

password: password

## 九、Telegraf采集端安装

### 1、下载：

wget https://dl.influxdata.com/telegraf/releases/telegraf-1.6.3-1.x86\_64.rpm

### 2、安装

rpm –ivh telegraf-1.6.3-1.x86\_64.rpm

### 3、配置

/etc/telegraf/telegraf.conf

urls = ["http://127.0.0.1:8086"]

database = "telegraf"

*# 跳过创建数据库*

skip\_database\_creation = false

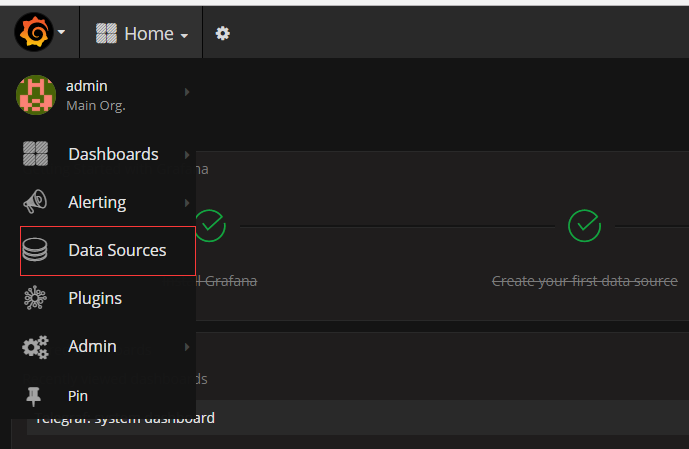
*# 超时时间*

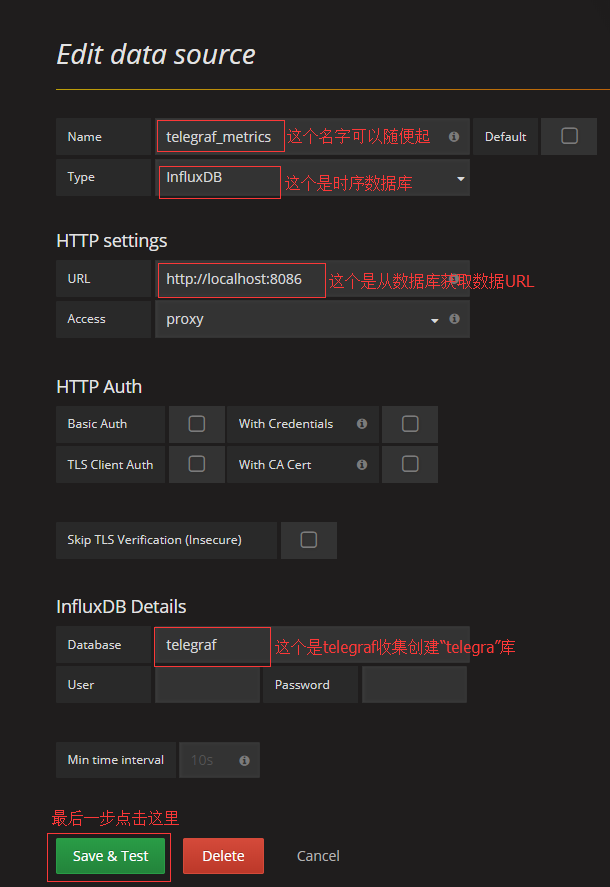
timeout = "5s"

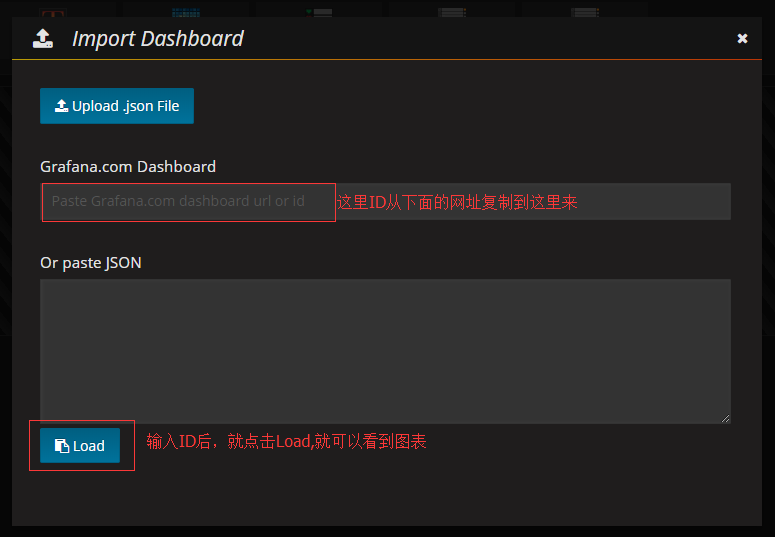
### 4、启动

systemctl start|status|restart|stop telegraf

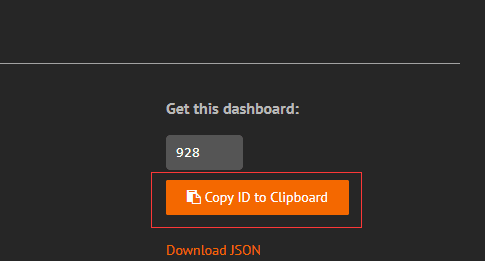
## 十、grafana整合显示







然后从别人编辑好面板复制[https://grafana.com/dashboards/928](https://note.youdao.com/)ID到上面





## 遇到的问题：

1、卸载包 rpm -qa | grep jdk

rpm -e --nodeps 要卸载的软件包

1、编译安装git提示：Can't locate ExtUtils/MakeMaker.pm in @INC

解决：

yum install perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker

2、

git 网上拉代码遇到 error: RPC failed; result=18, HTTP code = 200

git config –list

解决：

终端输入命令：

## modify in commandline

git config --global http.postBuffer 524288000

## check config list

git config –list

3、

can't load package: package mygrafana: build constraints exclude all Go files in /usr/local/gohome/src/mygrafana、

缺少gcc

4、[root@localhost grafana]# npm install -g yarn

npm ERR! code ENOTFOUND

npm ERR! errno ENOTFOUND

npm ERR! network request to https://registry.npmjs.org/yarn failed, reason: getaddrinfo ENOTFOUND registry.npmjs.org registry.npmjs.org:443

npm ERR! network This is a problem related to network connectivity.

npm ERR! network In most cases you are behind a proxy or have bad network settings.

npm ERR! network

npm ERR! network If you are behind a proxy, please make sure that the

npm ERR! network 'proxy' config is set properly. See: 'npm help config'

npm ERR! A complete log of this run can be found in:

npm ERR! /root/.npm/\_logs/2019-04-04T02\_26\_41\_660Z-debug.log

解决：npm config rm proxy

npm config rm https-proxy

5、

error An unexpected error occurred: "https://registry.yarnpkg.com/babel-runtime/-/babel-runtime-6.26.0.tgz: ETIMEDOUT".

info If you think this is a bug, please open a bug report with the information provided in "/usr/local/gohome/src/github.com/grafana/grafana/yarn-error.log".

yarn config set sass-binary-site http://npm.taobao.org/mirrors/node-sass

npm config rm proxy

npm config rm https-proxy

make: g++：命令未找到