ClickHouse

监控及备份

版本：V1.0

# 第1章 ClickHouse监控概述

ClickHouse 运行时会将一些个自身的运行状态记录到众多系统表中( system.\*)。所以我们对于 CH 自身的一些运行指标的监控数据，也主要来自这些系统表。

但是直接查询这些系统表会有一些不足之处：

* 这种方式太过底层，不够直观，我们还需要在此之上实现可视化展示；
* 系统表只记录了CH自己的运行指标，有些时候我们需要外部系统的指标进行关联分析，例如 ZooKeeper、服务器 CPU、IO 等等。

现在Prometheus + Grafana的组合比较流行，安装简单易上手，可以集成很多框架，包括服务器的负载, 其中 Prometheus 负责收集各类系统的运行指标; Grafana 负责可视化的部分。

ClickHouse 从 v20.1.2.4 开始，内置了对接 Prometheus 的功能，配置的方式也很简单,可以将其作为 Prometheus 的 Endpoint 服务，从而自动的将 metrics、events 和 asynchronous\_metrics 三张系统的表的数据发送给 Prometheus。

# 第2章 Prometheus&Grafana的安装

Prometheus下载地址：<https://prometheus.io/download/>

Grafana下载地址：<https://grafana.com/grafana/download>

## 2.1 安装Prometheus

Prometheus基于Golang编写，编译后的软件包，不依赖于任何的第三方依赖。只需要下载对应平台的二进制包，解压并且添加基本的配置即可正常启动Prometheus Server。

### 2.1.1 上传安装包

上传prometheus-2.26.0.linux-amd64.tar.gz到虚拟机的/opt/software目录

### 2.1.2 解压安装包

（1）解压到/opt/module目录下

[atguigu@hadoop1 software]$ tar -zxvf prometheus-2.26.0.linux-amd64.tar.gz -C /opt/module

（2）修改目录名

[atguigu@hadoop1 ~] cd /opt/module

[atguigu@hadoop1 module] mv prometheus-2.26.0.linux-amd64 prometheus-2.26.0

### 2.1.3 修改配置文件 prometheus.yml

[atguigu@hadoop1 prometheus-2.26.0]$ vim prometheus.yml

在scrape\_configs配置项下添加配置：

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['hadoop1:9090']

#添加ClickHouse监控配置

- job\_name: clickhouse-1

static\_configs:

- targets: ['hadoop1:9363']

配置说明：

**1、global配置块：**控制Prometheus服务器的全局配置

* scrape\_interval：配置拉取数据的时间间隔，默认为1分钟。
* evaluation\_interval：规则验证（生成alert）的时间间隔，默认为1分钟。

**2、rule\_files配置块：**规则配置文件

**3、scrape\_configs配置块：**配置采集目标相关， prometheus监视的目标。Prometheus自身的运行信息可以通过HTTP访问，所以Prometheus可以监控自己的运行数据。

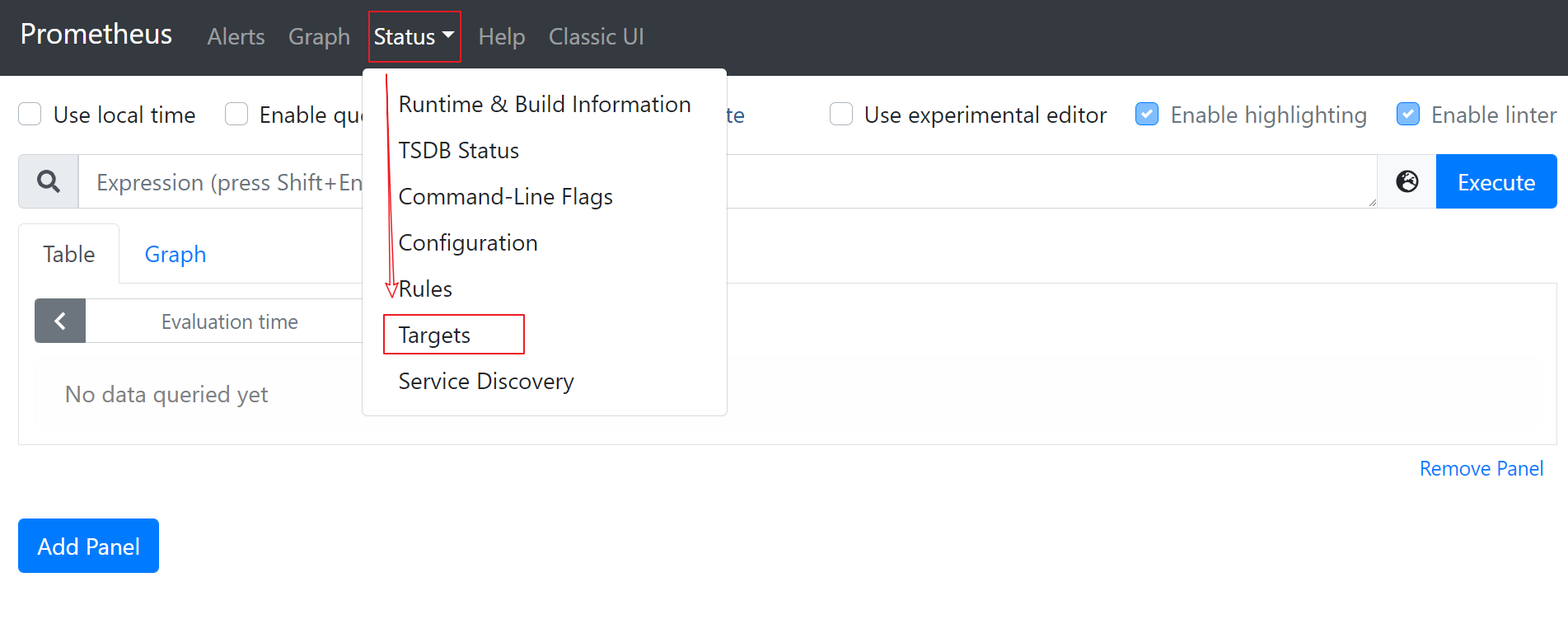
* job\_name：监控作业的名称
* static\_configs：表示静态目标配置，就是固定从某个target拉取数据
* targets：指定监控的目标，其实就是从哪儿拉取数据。Prometheus会从http://hadoop1:9090/metrics上拉取数据。

***Prometheus是可以在运行时自动加载配置的。启动时需要添加：--web.enable-lifecycle***

### 2.1.4 启动Prometheus Server

[atguigu@hadoop1 prometheus-2.26.0]$ nohup ./prometheus --config.file=prometheus.yml > ./prometheus.log 2>&1 &

* 浏览器输入：<http://hadoop1:9090/>
* 点击 Status，选中Targets：



*prometheus是up状态，表示安装启动成功：*

## 2.2 Grafana安装

### 2.2.1 上传并解压

（1）将grafana-7.5.2.linux-amd64.tar.gz上传至/opt/software/目录下，解压：

[atguigu@hadoop1 software]$ tar -zxvf grafana-7.5.2.linux-amd64.tar.gz -C /opt/module/

（2）更改名字：

[atguigu@hadoop1 module]$ mv grafana-7.5.2.linux-amd64 grafana-7.5.2

### 2.2.2 启动Grafana

[atguigu@hadoop1 grafana-7.5.2]$ nohup ./bin/grafana-server web > ./grafana.log 2>&1 &

打开web：<http://hadoop1:3000>,默认用户名和密码：admin

# 第3章 ClickHouse配置

## 3.1 修改配置文件

编辑/etc/clickhouse-server/config.xml，打开如下配置：

<prometheus>

<endpoint>/metrics</endpoint>

<port>9363</port>

<metrics>true</metrics>

<events>true</events>

<asynchronous\_metrics>true</asynchronous\_metrics>

<status\_info>true</status\_info>

</prometheus>

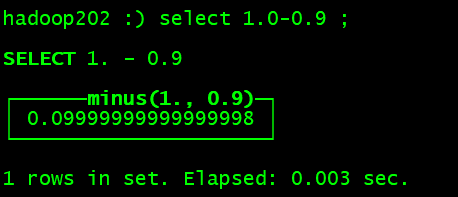
如果有多个CH节点，分发配置。

## 3.2 重启ClickHouse

sudo clickhouse restart

Float64 – double

建议尽可能以整数形式存储数据。例如，将固定精度的数字转换为整数值，如时间用毫秒为单位表示，因为浮点型进行计算时可能引起四舍五入的误差。



使用场景：一般数据值比较小，不涉及大量的统计计算，精度要求不高的时候。比如保存商品的重量。

## 3.3 访问Web查看

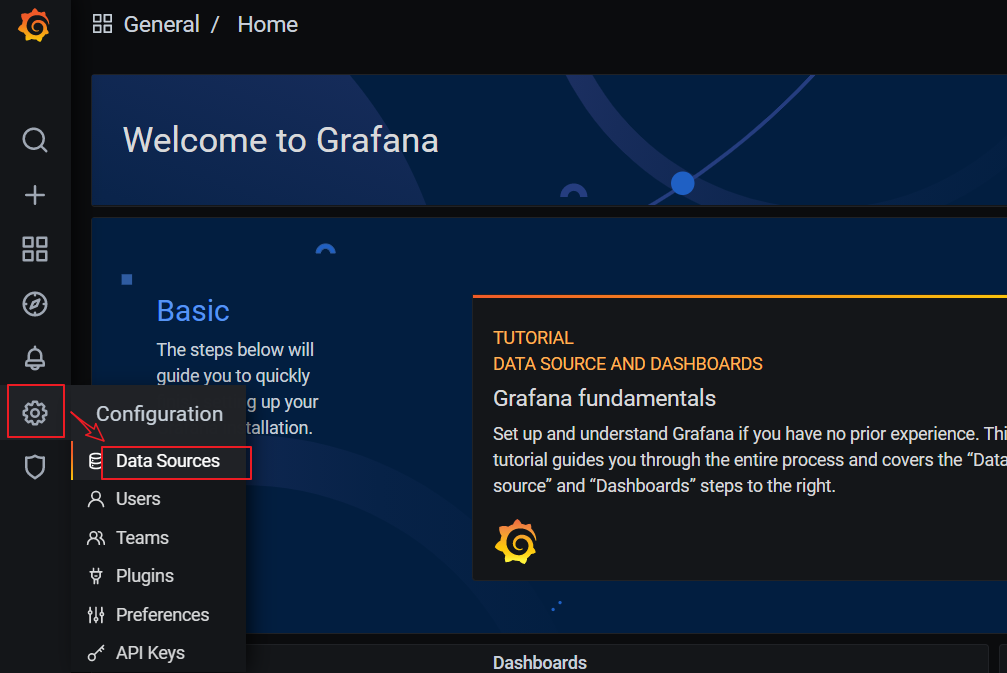
浏览器打开: <http://hadoop1:9363/metrics>

看到信息说明ClickHouse开启Metrics服务成功。

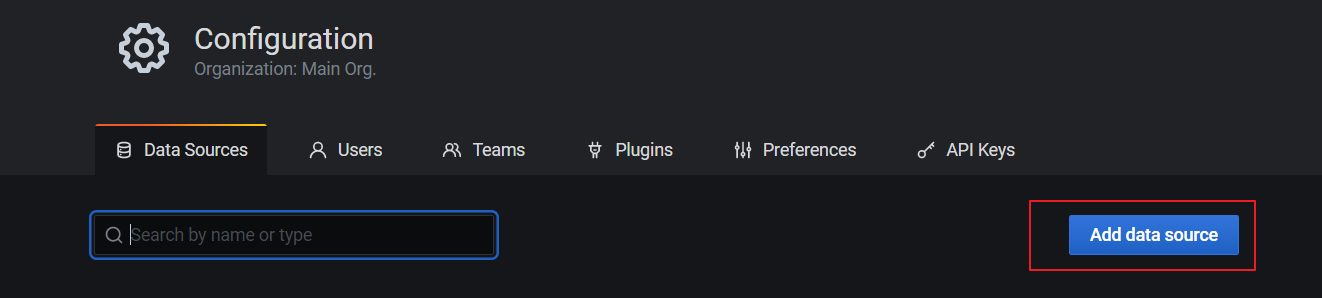
# 第4章 Grafana集成Prometheus

## 4.1 添加数据源Prometheus

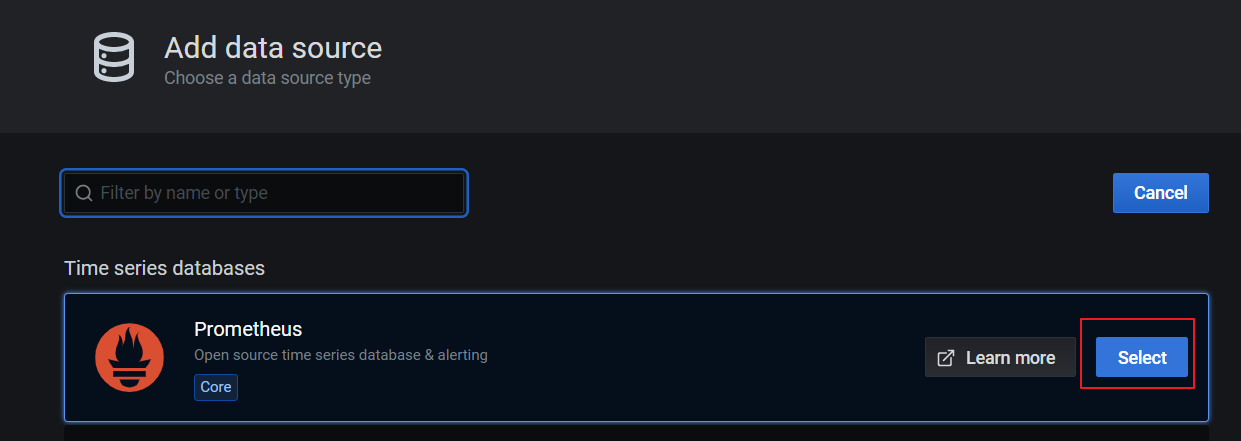
（1）点击配置，点击Data Sources：



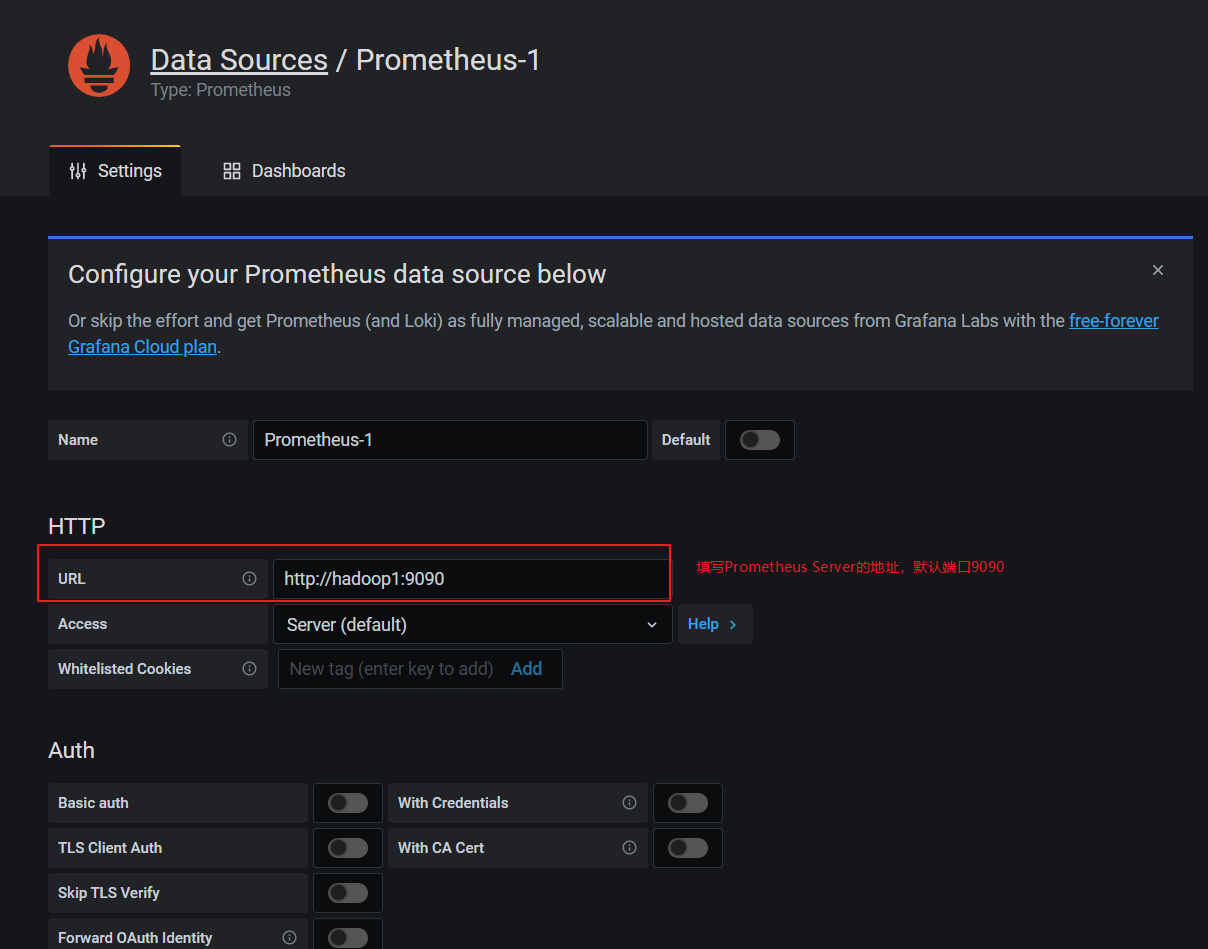
（2）点击添加按钮：



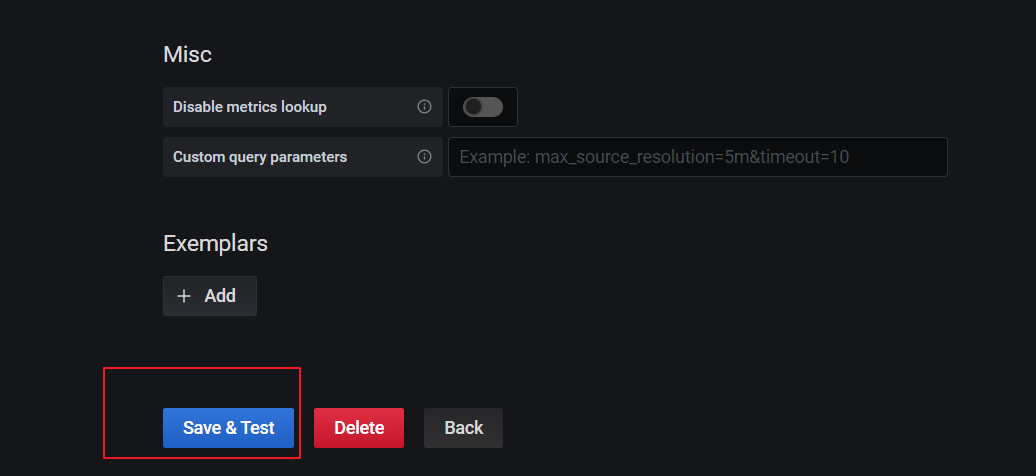
（3）找到Prometheus，点击Select



（4）配置Prometheus Server地址：



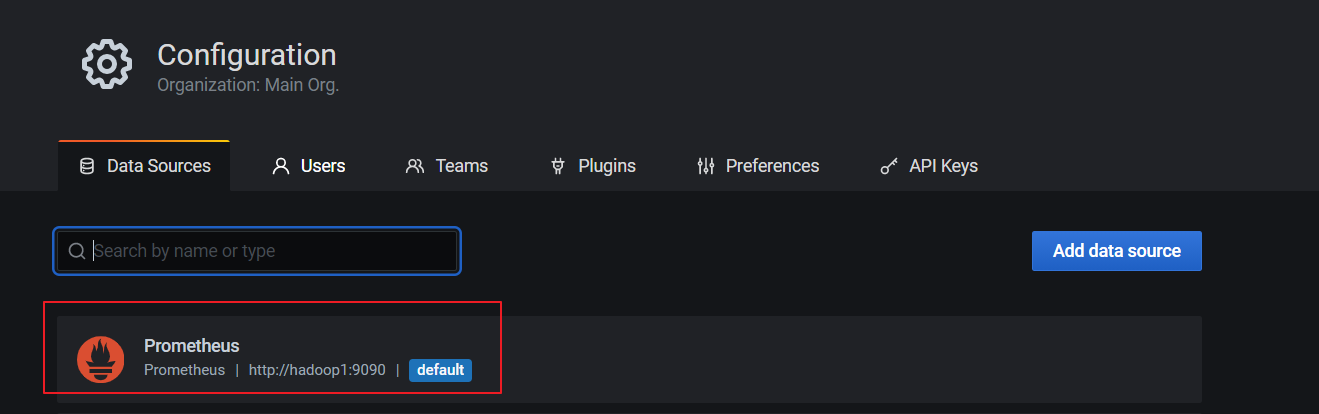
（5）点击下方的Save&Test：



（6）出现绿色的提示框，表示与Prometheus正常联通：



（7）点击Back返回即可，可以看到Data Sources页面，出现了添加的Prometheus:

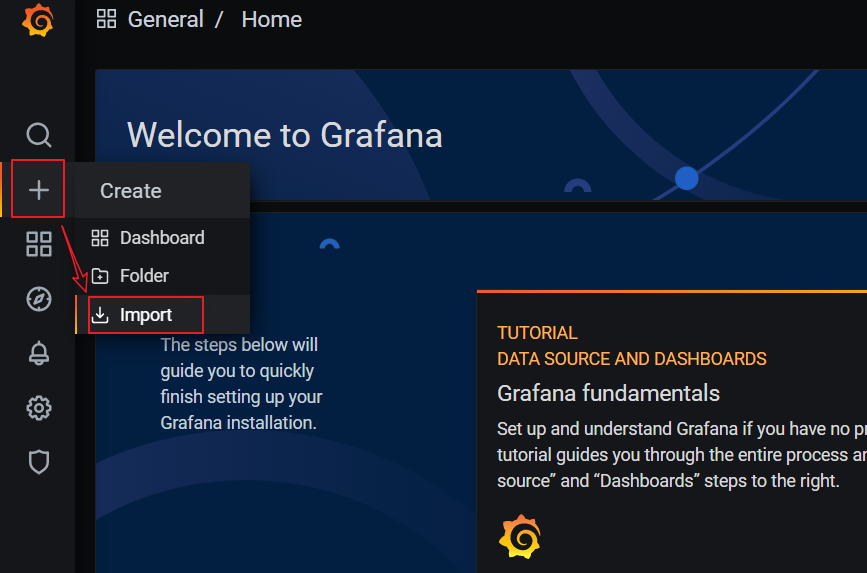


## 4.2 添加监控

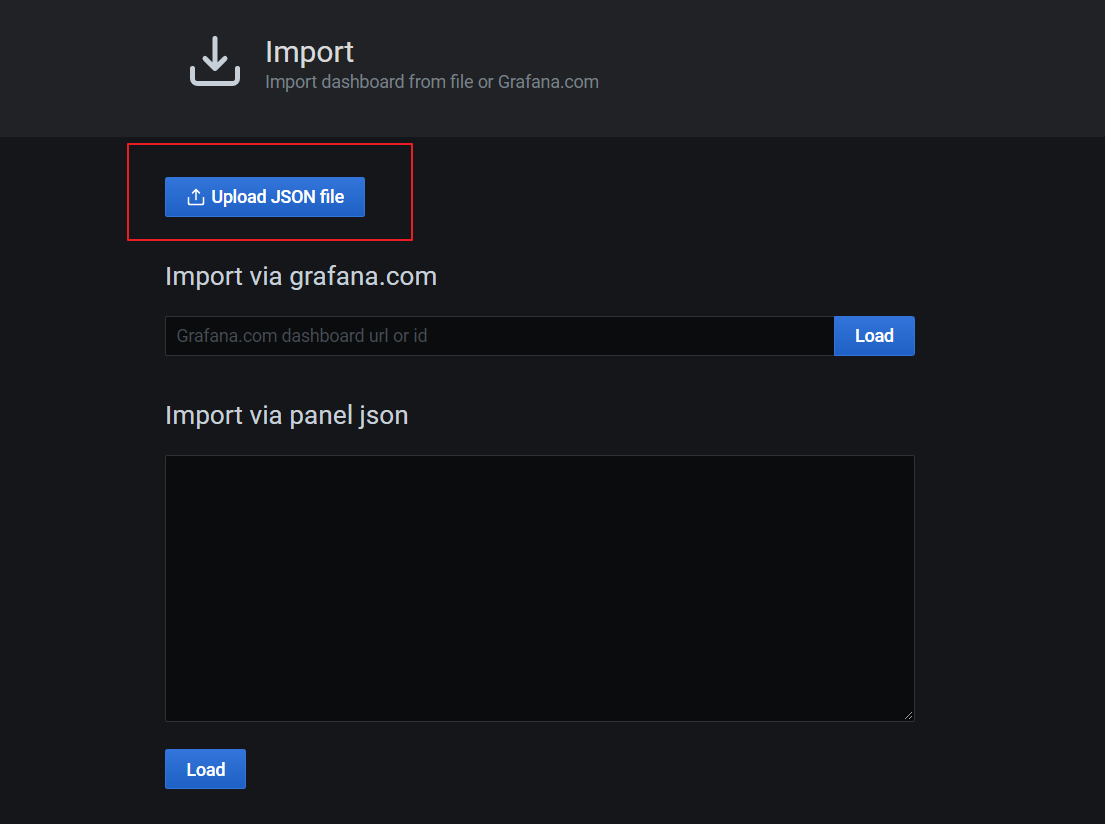
手动一个个添加Dashboard比较繁琐，Grafana社区鼓励用户分享Dashboard，通过<https://grafana.com/dashboards>网站，可以找到大量可直接使用的Dashboard模板。

Grafana中所有的Dashboard通过JSON进行共享，下载并且导入这些JSON文件，就可以直接使用这些已经定义好的Dashboard。

（1）点击左侧 ”+”号，选择import：



（2）上传JSON文件：



# 第5章 备份及恢复

官网：<https://clickhouse.tech/docs/en/operations/backup/>

## 5.1 手动实现备份及恢复

ClickHouse允许使用 ALTER TABLE ... FREEZE PARTITION ... 查询以创建表分区的本地副本。 这是利用硬链接(hardlink)到 /var/lib/clickhouse/shadow/ 文件夹中实现的，所以它通常不会因为旧数据而占用额外的磁盘空间。 创建的文件副本不由ClickHouse服务器处理，所以不需要任何额外的外部系统就有一个简单的备份。防止硬件问题，最好将它们远程复制到另一个位置，然后删除本地副本。

### 5.1.1 创建备份路径

（1）创建用于存放备份数据的目录shadow

sudo mkdir -p /var/lib/clickhouse/shadow/

如果目录存在，先清空目录下的数据

### 5.1.2 执行备份命令

echo -n 'alter table t\_order\_mt freeze' | clickhouse-client

### 5.1.3 将备份数据保存到其他路径

#创建备份存储路径

sudo mkdir -p /var/lib/clickhouse/backup/

#拷贝数据到备份路径

sudo cp -r /var/lib/clickhouse/shadow/ /var/lib/clickhouse/backup/my-backup-name

#为下次备份准备，删除shadow下的数据

sudo rm -rf /var/lib/clickhouse/shadow/\*

### 5.1.4 恢复数据

（1）模拟删除备份过的表

echo ' drop table t\_order\_mt ' | clickhouse-client

（2）重新创建表

cat events.sql | clickhouse-client

（3）将备份复制到detached目录

sudo cp -rl backup/my-backup-name/1/store/cb1/cb176503-cd88-4ea8-8b17-6503cd888ea8/\* data/default/t\_order\_mt/detached/

ClickHouse使用文件系统硬链接来实现即时备份，而不会导致ClickHouse服务停机（或锁定）。这些硬链接可以进一步用于有效的备份存储。在支持硬链接的文件系统（例如本地文件系统或NFS）上，将cp与-l标志一起使用（或将rsync与–hard-links和–numeric-ids标志一起使用）以避免复制数据。

*注意：仅拷贝分区目录，注意目录所属的用户要是clickhouse*

（4）执行attach

echo 'alter table t\_order\_mt attach partition 20200601' | clickhouse-client

（5）查看数据

echo 'select count() from t\_order\_mt' | clickhouse-client

## 5.2 使用clickhouse-backup

上面的过程，我们可以使用Clickhouse的备份工具clickhouse-backup帮我们自动化实现。

工具地址：<https://github.com/AlexAkulov/clickhouse-backup/>

### 5.2.1 上传并安装

将clickhouse-backup-1.0.0-1.x86\_64.rpm上传至/opt/software/目录下，安装：

[atguigu@hadoop1 software]$ sudo rpm -ivh clickhouse-backup-1.0.0-1.x86\_64.rpm

### 5.2.2 配置文件

[atguigu@hadoop1 ~]$ cat /etc/clickhouse-backup/config.yml

### 5.2.3 创建备份

（1）查看可用命令

[atguigu@hadoop1 ~]$ clickhouse-backup help

（2）显示要备份的表

[atguigu@hadoop1 ~]$ clickhouse-backup tables

（3）创建备份

[atguigu@hadoop1 ~]$ sudo clickhouse-backup create

（4）查看现有的本地备份

[atguigu@hadoop1 ~]$ sudo clickhouse-backup list

备份存储在中/var/lib/clickhouse/backup/BACKUPNAME。备份名称默认为时间戳，但是可以选择使用–name标志指定备份名称。备份包含两个目录：一个“metadata”目录，其中包含重新创建架构所需的DDL SQL语句；以及一个“shadow”目录，其中包含作为ALTER TABLE ... FREEZE操作结果的数据。

### 5.2.4 从备份恢复数据

（1）模拟删除备份过的表

echo 'drop table t\_order\_rmt' | clickhouse-client

（2）从备份还原

sudo clickhouse-backup restore 2021-07-25T23-14-50

--schema参数：只还原表结构。

--data参数：只还原数据。

--table参数：备份（或还原）特定表。也可以使用一个正则表达式，例如，针对特定的数据库：--table=dbname.\*。

### 5.2.5 其他说明

（1）API文档：[https : //github.com/AlexAkulov/clickhouse-backup#api](https%20:%20/github.com/AlexAkulov/clickhouse-backup" \l "api)

（2）注意事项：切勿更改文件夹/var/lib/clickhouse/backup的权限，可能会导致数据损坏。

（3）远程备份

* 较新版本才支持，需要设置config里的s3相关配置
* 上传到远程存储：sudo clickhouse-backup upload xxxx
* 从远程存储下载：sudo clickhouse-backup download xxxx
* 保存周期： backups\_to\_keep\_local，本地保存周期，单位天

backups\_to\_keep\_remote，远程存储保存周期，单位天

0均表示不删除