

时序数据库 CTSDB

最佳实践



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

最佳实践

数据表定义推荐

接入 Grafana 服务

接入 Telegraf 服务

最佳实践

数据表定义推荐

最近更新时间：2024-05-16 17:00:42

在设计数据表时，有一些需要遵循的准则和需要避免的误区，详情请参见官方文档 [InfluxDB schema design and data layout](#)。本文也给出创建数据表的建议，以帮助您在创建高效的时序数据库。

数据库表命名

在定义 InfluxDB 数据表名称时，应该使用有意义的名称，以便可以轻松识别和理解表的内容。以下是一些有关定义有意义名称的建议：

- 使用描述性的名称，例如，如果您正在存储传感器数据，则可以使用传感器名称作为数据表名称。
- 避免使用缩写或简写词，因为这可能会导致混淆和不必要的困惑。
- 使用小写字母和连字符（-）来分隔单词，这样可以使名称更易于阅读和理解。
- 在命名数据表时，尽量避免使用特殊字符，因为这可能会导致查询和其他操作出现问题。
- 尽量使用短而简洁的名称，以避免在查询和其他操作中输入过多的字符。

Tag 与 Field

- Tag 通常用于描述和标识数据，例如设备 ID、网络地址和地理位置等。它们有助于提高查询性能和优化系统资源的开销。Tag 值在查询时用于过滤和分组（如 Group by），因此在选择 Tag 值时应避免不必要的冗余和嵌套。
- Field 通常用于描述和记录数据的趋势和变化，例如温度、电量、流量和速度等。Field 值可用于进行数学计算和统计分析，例如求平均值、最大值和最小值等。与 Tag 相比，由于 Field 值需要进行聚合和计算，Field 值的存储和查询开销较高，因此需要更多的 CPU 和内存资源支持。

将数据存储于 Tag 中

在 InfluxDB 中，Tag 是用于标识和描述数据的元数据。将数据存储于 Tag 中，可以提高数据库查询性能，更有效地管理和组织数据。以下是有关将数据存储于 Tag 中的建议：

- 存储常用的元数据：如果某个数据是与其他数据相关的元数据，例如传感器名称、位置信息、数据源等，则将其存储于 Tag 中，这样可方便查询和过滤数据。
- GROUP BY() 中使用：如果您计划在 GROUP BY() 语句中使用某个数据，则将其存储于 Tag 中。在 GROUP BY() 中指定 Tag 名称，可以将查询结果按照 Tag 的值进行分组。
- 快速查询：如果您需要对某个数据进行快速查询，则将其存储于 Tag 中。由于 Tag 被索引，在 Tag 上进行查询比在 Field 上进行查询效率更高。

将数据存储于 Field 中

在 InfluxDB 中，Field 是用于存储数据的元素。将数据存储于 Field 中可以方便进行 InfluxQL 函数操作，并且可以将其解释为除字符串之外的其他类型。以下是一些有关将数据存储于 Field 中的建议：

- 存储数值数据：如果您需要对数据进行数值计算或比较，则将其存储于 Field 中，这样可以方便地使用 InfluxQL 函数对数据进行操作。
- 存储时间戳数据：如果您需要对数据进行时间序列分析，则将其存储于 Field 中。InfluxDB 支持时间戳类型的 Field，并提供了一些用于时间序列分析的函数。
- 避免存储重复的数据：如果某个数据已经存储于 Tag 中，则不需要再将其存储于 Field 中。这样可以避免数据重复，并使查询效率更高。
- 使用正确的数据类型：在定义 Field 时，应该使用正确的数据类型。例如，如果您正在存储数值数据，则应该使用数值类型的 Field，而不是字符串类型的 Field。

数据 Field 分组

在 InfluxDB 中，Field 是用于存储数据的元素。在定义数据表时，将 Field 分组为相关的测量类型，有助于组织和管理数据，并使查询效率更高。以下是一些有关将 Field 分组的建议：

- 根据测量类型分组：将具有相同测量类型的 Field 分组到一起。例如，如果您正在存储传感器数据，则可以将温度、湿度、压力等测量类型分别分组。
- 使用连字符分隔单词：在命名 Field 时，使用连字符（-）而不是下划线（_）来分隔单词。因为在 InfluxDB 中下划线用于分隔 Field 和 Tag，使用连字符（-）可以避免混淆。
- 组织数据表结构：在定义数据表时，考虑将不同测量类型的 Field 分组为不同的表结构，这样有助于更好地组织和管理数据，并使查询效率更高。

- 使用通配符进行查询：在查询时，可以使用通配符来查询相同测量类型的所有 Field。例如，如果您想要查询所有温度字段，则可以使用通配符 “*” 来查询所有以 “temp” 结尾的 Field。

接入 Grafana 服务

最近更新時間：2024-08-30 10:43:52

Grafana 是一個跨平台的開源的可视化分析工具，是目前網絡架構和應用分析中最流行的時序數據展示工具，主要用於大規模指標數據的可视化展示。更多信息，請參見 [Grafana 服務](#)。將時序數據庫 CTSDB 3.0 版接入 Grafana，您可以利用 Grafana 的豐富易用的可视化工具更好地監控和分析來自時序數據庫 CTSDB 3.0 版的數據。

前置條件

1. 在接入 Grafana 之前，您需要先購買時序數據庫 CTSDB 3.0 版實例，並配置好用戶賬號與數據庫信息，同時寫入一定量的數據。具體操作，請參見 [新建數據庫實例](#) 與 [連接並寫入數據](#)。
2. 下載並安裝 Grafana 服務。
 - （推薦）購買腾讯云基於社區廣受歡迎的開源可视化項目 Grafana 開發的 [Grafana 服務（Tencent Cloud Managed Service for Grafana, TCMG）](#)。具體操作，請參見 [Grafana 服務創建實例](#)。

注意：

- 腾讯云 Grafana 服務（Tencent Cloud Managed Service for Grafana, TCMG）請務必與時序數據庫 CTSDB 3.0 版實例在同一个 VPC 網絡，否則無法連通。
- 在腾讯云 Grafana 安全組中配置出站規則，需要把時序數據庫 CTSDB 3.0 版的 IP 及端口添加到出站規則中。在時序數據庫 CTSDB 3.0 版安全組中配置入站規則，把 Grafana 的 IP 及端口添加到入站規則中。

- 在腾讯云 CVM 上的 Windows 操作系統瀏覽器訪問 [Grafana 官方網站的下载地址](#)，根據自己的系統版本和配置，下載對應的安裝包進行安裝即可。

注意：

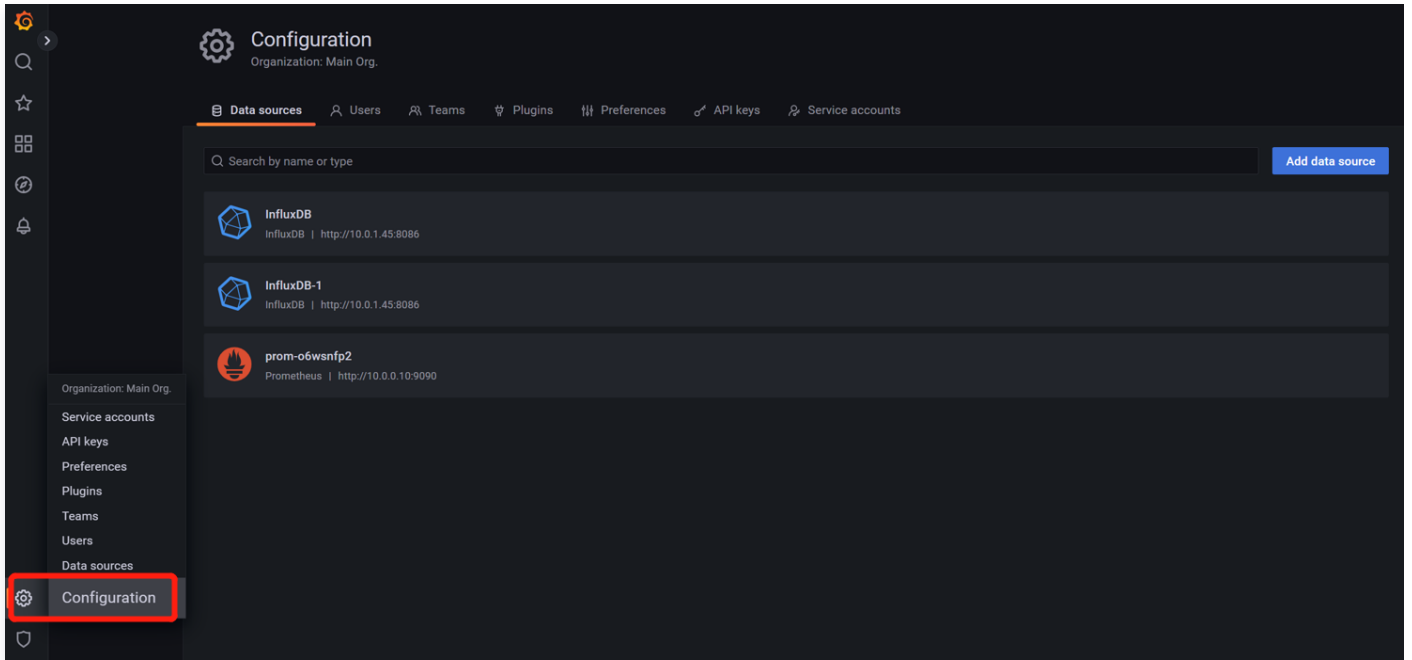
- 官方 Grafana v8.x 及之前的版本均已接入驗證，當前因時序數據庫 CTSDB 3.0 版暫時未開放 `show retention policies` 接口，Grafana 添加數據源時使用 Show RP 對數據源進行檢查，導致在添加數據源時提示 `Network Error: Bad Request (400)` 錯誤。您可以忽略此條信息，數據源是可以正常添加並使用的。
- 腾讯云 CVM 請務必與時序數據庫 CTSDB 3.0 版實例在同一个 VPC 網絡。
- 在腾讯云 CVM 安全組中配置出站規則，需要把時序數據庫 CTSDB 3.0 版的 IP 及端口添加到出站規則中。在時序數據庫 CTSDB 3.0 版安全組中配置入站規則，把 CVM 的 IP 及端口添加到入站規則中。

接入步驟

以下介紹接入腾讯云 Grafana 服務的接入步驟，開源的 Grafana 服務配置過程與其類似。

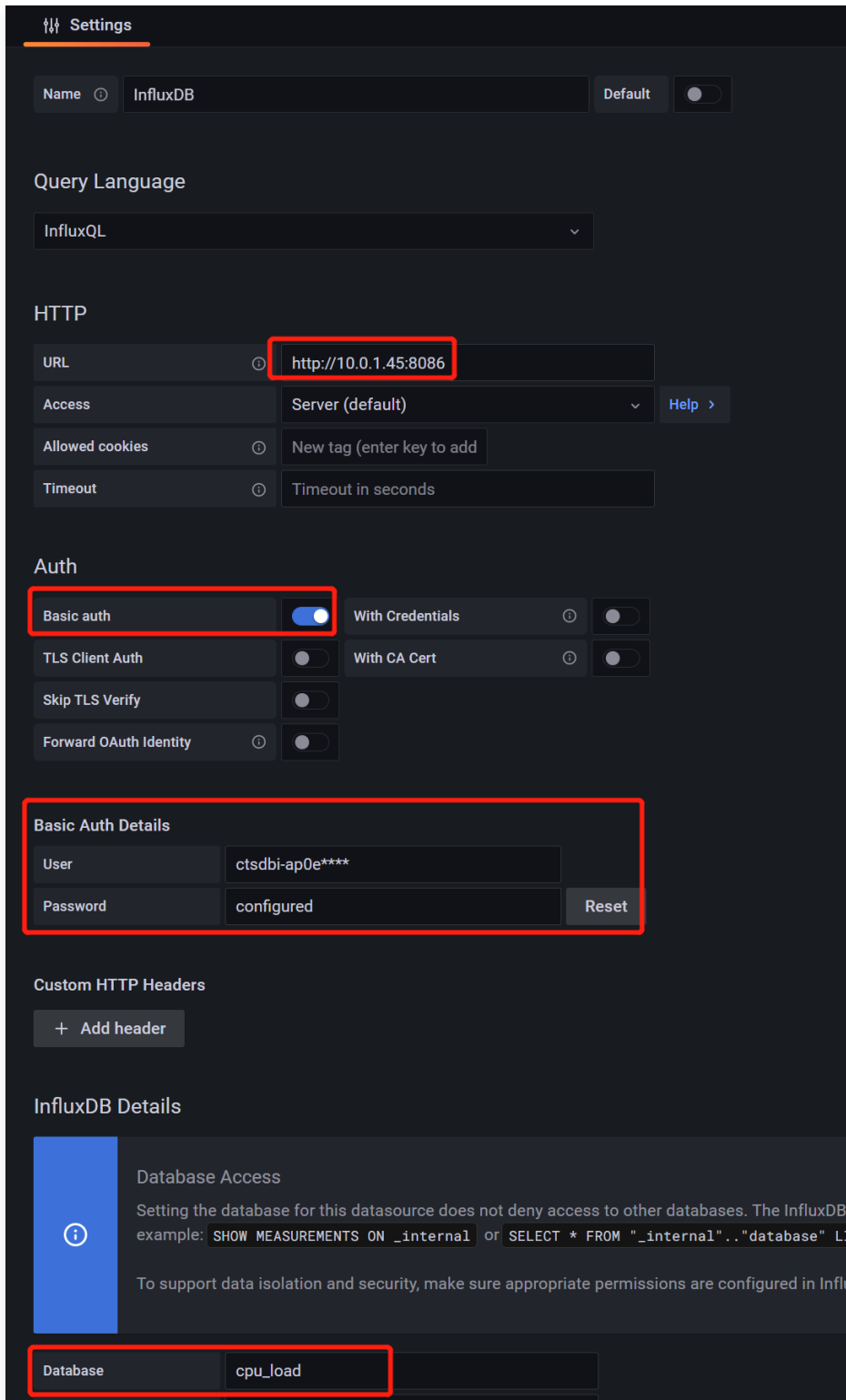
1. 登錄 [Grafana 服務控制台](#)。
2. 在實例列表中，找到已創建的 Grafana 實例，單擊實例 ID，進入 Grafana 服務基本信息頁面。
3. 在基本信息頁面的網絡區域，單擊 打開外網地址。
4. 單擊基本信息區域的訪問地址，登錄 Grafana 服務。

5. 在左下角，单击 ，并选择 Configuration，进入 Data sources 页面，如下图所示：



6. 在搜索框输入 influxDB，找到 InfluxDB 数据库，配置数据源参数如下：

界面参数	子参数	参数含义	示例
URL	-	配置为时序数据库 CTSDB 3.0 版的内网访问地址。	例如：内网地址为 10.0.1.45，端口为：8086。 则输入：http://10.0.1.45:8086。
Auth	Basic auth	配置访问时序数据库 CTSDB 3.0 版是否需认证。	开启
	Basic Auth Details	该参数仅开启Basic auth 才显示，输入访问时序数据库 CTSDB 3.0 版的用户名	ctsdbi-ap0e****
	Password	输入访问时序数据库 CTSDB 3.0 版的用户名对应的密码。	test@123
InfluxDB Details	Database	配置数据源的数据库名.请务必写入数据，否则测试将失败。	cpudata



7. 配置完成，单击下方 **Save & test**，保存配置并测试连通性。提示 `datasource is working. 1 measurements found` 信息，说明接入成功。
8. 单击左上方的 ，并选择 **New dashboard**，创建图形，Data source 选择为已连通的数据源，如下所示。更多绘制图标的操作，请参见 [开始使用 Grafana 和 InfluxDB](#)。

← New dashboard / Edit Panel Table view

Panel Title

11:45 12:00 12:15 12:30 12:45 13:00 13:15 13:30 13:45 14:00 14:15 14:30 14:45 15:00 15:15 15:30 15:45 16:00 16:15 16:30 16:45 17:00 17:15 17:30

A-series

Query 1 Transform 0 Alert 0

Data source **InfluxDB** Query options MD = auto = 1234 Interval = 20s

FROM	default	select measurement	WHERE	+
SELECT	field(value)	mean()	+	
GROUP BY	time(\$interval)	fill(null)	+	
TIMEZONE	(optional)	ORDER BY TIME	ascending	▼
LIMIT	(optional)	SLIMIT	(optional)	
FORMAT AS	Time series	ALIAS	Naming pattern	

接入 Telegraf 服务

最近更新时间：2024-08-30 10:43:52

Telegraf 是 InfluxData 旗下的一款功能强大的数据采集工具。它的主要功能是采集各种不同服务的信息，包括但不限于系统 CPU 使用率、内存占用情况以及一些常用服务如 Redis 和 Nginx 等的运行状况。Telegraf 与时序数据库 InfluxDB 版进行整合，将采集到的数据保存在 InfluxDB 中，以便帮助用户获取各种服务的信息并进行实时监控，为用户提供更好的数据采集和分析体验。更多信息，请参见官网 [Telegraf](#)。

前置条件

1. 在接入 Telegraf 之前，您需要先购买时序数据库 CTSDB 3.0 版实例，并配置好用户账号，创建好数据库。具体操作，请参见 [新建数据库实例与连接并写入数据](#)。
2. 登录 [云服务器控制台](#)，购买 CVM 服务器，可 [快速配置 Linux 云服务器](#) 或 [快速配置 Windows 云服务器](#)。
3. 腾讯云 CVM 请务必与时序数据库 InfluxDB 版实例在同一个 VPC 网络。
4. 在腾讯云 CVM 安全组中配置出站规则，需要把时序数据库 CTSDB 3.0版的 IP 及端口添加到出站规则中。在时序数据库 CTSDB 3.0版安全组中配置入站规则，把 CVM 的 IP 及端口添加到入站规则中。操作详情，可参见 [安全组](#)。

接入步骤

如下操作以 Linux 操作系统为例，其他操作系统，请参见 [Install Telegraf](#) 选择对应的安装包并下载进行配置。

1. 登录 [云服务器控制台](#)，在实例列表，找到已购买的服务器实例，单击其操作列的登录。
2. 以 Center OS 为例，添加 InfluxDB 软件包的 Yum 仓库源，并安装，默认安装在 etc 目录。具体操作，请参见官网 [安装 Telegraf](#)。如下图所示为以 Center OS 为例的安装过程提示信息。

```
[root@M-3-2-centos ~]# cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo
> [influxdb]
> name = InfluxData Repository - Stable
> baseurl = https://repos.influxdata.com/stable/$basearch/main
> enabled = 1
> gpgcheck = 1
> gpgkey = https://repos.influxdata.com/influxdata-archive_compat.key
> EOF
[influxdb]
name = InfluxData Repository - Stable
baseurl = https://repos.influxdata.com/stable/$basearch/main
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdata-archive_compat.key
[root@M-3-2-centos ~]# yum install telegraf
Invalid configuration value: failovermethod-priority in /etc/yum.repos.d/CentOS-Epel.repo; Configuration: OptionBinding with id "failovermethod" does not exist
InfluxData Repository - Stable                               31 kB/s | 51 kB   00:01
Dependencies resolved.
-----
Package                Architecture      Version           Repository         Size
-----
Installing:
telegraf                x86_64            1.26.3-1         influxdb           47 M

Transaction Summary
-----
Install 1 Package
```

执行成功信息如下图所示：

```
Downloading Packages:
telegraf-1.26.3-1.x86_64.rpm                               2.2 MB/s | 47 MB   00:21
-----
Total                                                    2.2 MB/s | 47 MB   00:21
InfluxData Repository - Stable                          751 B/s | 1.6 kB   00:02
Importing GPG key 0x7DF8B07E:
  Userid   : "InfluxData Package Signing Key <support@influxdata.com>"
  Fingerprint: 9D53 9D90 D332 8DC7 D6C8 D389 D8FF 8E1F 7DF8 B07E
  From     : https://repos.influxdata.com/influxdata-archive_compat.key
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                     1/1
  Running scriptlet: telegraf-1.26.3-1.x86_64         1/1
  Installing     : telegraf-1.26.3-1.x86_64         1/1
  Running scriptlet: telegraf-1.26.3-1.x86_64         1/1
Created symlink /etc/systemd/system/multi-user.target.wants/telegraf.service → /usr/lib/systemd/system/telegraf.service.

  Verifying      : telegraf-1.26.3-1.x86_64         1/1

Installed:
  telegraf-1.26.3-1.x86_64

Complete!
```

3. 在正式启动之前，需要先进行相关配置，指定 Telegraf 对哪些数据进行采集，配置数据存储的地址等。执行 `cd etc` 命令，进入 `etc` 目录下，执行如下命令，生成 Telegraf 配置文件。

```
cd etc
telegraf config > telegraf.conf
```

4. 使用 `vi telegraf.conf` 打开文件，找到 `[[outputs.influxdb]]`，配置数据输出的时序数据库地址。如下所示，`urls` 配置为时序数据库 CTSDB 3.0 版的内网地址与端口；`database` 为数据库库名；`username` 与 `password` 配置访问数据库用户名与密码。

```
[[outputs.influxdb]]
urls = ["http://10.1.98.53:8086"]
database = "replicaset"
skip_database_creation = true
exclude_database_tag = true
exclude_retention_policy_tag = true
content_encoding = "identity"
timeout = "5s"
username = "ctsdbi-s0u*****"
password = "*****"
```

Input 默认配置信息如下，可根据需求配置所需采集的数据。

```
# Read metrics about cpu usage
[[inputs.cpu]]
## Whether to report per-cpu stats or not
percpu = true
## Whether to report total system cpu stats or not
totalcpu = true
## If true, collect raw CPU time metrics
collect_cpu_time = false
## If true, compute and report the sum of all non-idle CPU states
report_active = false
## If true and the info is available then add core_id and physical_id tags
core_tags = false
```

5. 执行如下命令，启动 Telegraf。

```
telegraf -config /etc/telegraf.conf
```

提示信息如下所示:

```
[root@VM-3-2-centos etc]# telegraf -config /etc/telegraf.conf
2023-06-07T11:03:21Z I! Loading config: /etc/telegraf.conf
2023-06-07T11:03:21Z I! Starting Telegraf 1.26.3
2023-06-07T11:03:21Z I! Available plugins: 235 inputs, 9 aggregators, 27 processors, 22 parsers, 57 outputs, 2 secret-stores
2023-06-07T11:03:21Z I! Loaded inputs: cpu disk diskio mem net
2023-06-07T11:03:21Z I! Loaded aggregators:
2023-06-07T11:03:21Z I! Loaded processors:
2023-06-07T11:03:21Z I! Loaded secretstores:
2023-06-07T11:03:21Z I! Loaded outputs: influxdb
2023-06-07T11:03:21Z I! Tags enabled: host=VM-3-2-centos
2023-06-07T11:03:21Z I! [agent] Config: Interval:10s, Quiet:false, Hostname:"VM-3-2-centos", Flush Interval:10s
```

6. 连接时序数据库 CTSDB 3.0版的数据, 执行 `show measurements` , 确认数据已正常写入。

```
[root@VM-3-2-centos tmp]# curl --get http://10.0.1.45:8086/query \
> --user "ctsdbi:XXXXXXXXXX" \
> --data-urlencode 'db=cpu_data' \
> --data-urlencode "pretty=true" \
> --data-urlencode "q=show measurements"
```