

# Telegraf

## Installation

- [Install Telegraf | Telegraf 1.26 Documentation \(influxdata.com\)](#)

## RHEL

```
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

sudo yum install telegraf
```

## Ubuntu/Debian

```
curl -s https://repos.influxdata.com/influxdata-archive_compat.key > influxdata-archive_compat.key
echo '393e8779c89ac8d958f81f942f9ad7fb82a25e133faddaf92e15b16e6ac9ce4c influxdata-
archive_compat.key' | sha256sum -c && cat influxdata-archive_compat.key | gpg --dearmor | sudo tee
/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg > /dev/null
echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive_compat.gpg] https://repos.influxdata.com/debian
stable main' | sudo tee /etc/apt/sources.list.d/influxdata.list
sudo apt-get update && sudo apt-get install telegraf
```

## Configuration

```
telegraf config > telegraf.conf

# Using filter
```

```
telegraf --input-filter exec --output-filter influxdb_v2 config > /etc/telegraf/telegraf.conf
```

```
# Test for the configuration
```

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

## Custom systemd

```
cp /usr/lib/systemd/system/telegraf.service /etc/systemd/system/telegraf-db2.servic
```

telegraf-db2.service:

```
## [Unit]
```

```
EnvironmentFile=-/etc/default/telegraf-db2
```

```
## [Service]
```

```
ExecStart=/usr/bin/telegraf -config /etc/telegraf/telegraf-db2.conf $TELEGRAF_OPTS
```

Reload the daemon

```
systemctl list-unit-files --type service
```

```
systemctl daemon-reload
```

## Outputs.InfluxDB v1

```
#####  
#####
```

```
#                OUTPUT PLUGINS                #
```

```
#####
```

```
#####
```

```
# Configuration for sending metrics to InfluxDB
```

```
[[outputs.influxdb]]
```

```
  urls = ["http://influxdb.server.ip.addr:8086"]
```

```
  database = "db-name"
```

```
  timeout = "0s"
```

```
  username = "db-user"
```

```
  password = "db-pass"
```

## Outputs.InfluxDB v2

```
#####  
#####  
#                OUTPUT PLUGINS                #  
#####  
#####  
  
[[outputs.influxdb_v2]]  
  urls = ["http://influxdb.server.ip.addr:8086"]  
  token = "example-token"  
  organization = "example-org"  
  bucket = "example-bucket"
```

## Inputs.exec

data\_format = "influx"

???????

```
# Syntax for Line protocol  
<measurement>[,<tag_key>=<tag_value>[,<tag_key>=<tag_value>]]  
<field_key>=<field_value>[,<field_key>=<field_value>] [<timestamp>]  
                                     |                               |  
                                     <whitespace>                 <whitespace>  
  
# Example  
airsensors,location=bedroom,sensor_id=MI0201 temperature=19.1,humidity=85i,battery=78i  
1556813561098000000
```

- “
- ?? Timestamp ????????????? InfluxDB ??????(UTC)?
  - ?????[Line protocol | InfluxDB OSS v2 Documentation](https://docs.influxdata.com/influxdb/v2.0/line_protocol/)  
([influxdata.com](https://docs.influxdata.com/influxdb/v2.0/line_protocol/))
  - field\_value ??? Integer????? i?? String?????????
  - measurename, tag\_key, tag\_value, field\_key ??????????
  - ?????? measurename, field\_key, field\_value?
  - ?????????????????????? (\n) ?????
  - ??????????

## Plugins

- [Plugin directory | Telegraf Documentation \(influxdata.com\)](https://influxdata.com/docs/plugins/)

# Scripts

## Samples #1

```
#!/bin/bash
```

```
devname=(`lsblk| grep 'disk'|awk '{print $1}'`)
```

```
dirname=(`lsblk| grep 'disk'|awk '{if ($7=="") print "/";else print $7}'`)
```

#At that time, I wanted to store these directory names in dictionary format, and later changed to variable mode,  
shell Of [ ] { } \* @ \$Special characters will drive you crazy

```
#declare -A devdict
```

```
devnum=`expr ${#devname[@]} - 1`
```

```
for i in `seq 0 $devnum`;do
```

```
if [-z "${dirname[$i]}" ];then
```

```
eval ${devname[$i]}="/"
```

```
else
```

```
eval ${devname[$i]}="${dirname[$i]}"
```

```
fi
```

```
#devdict+=([${devname[$i]}]=${dirname[$i]})
```

```
done
```

```
#echo ${!devdict[*]}
```

```
#echo ${devdict[*]}
```

```
ioarray=`iostat -x | grep sd|awk '{print
```

```
"datadir=${!1}"@r="$4",w="$5",await="$10",svctm="$11",util="$12}'`
```

```
for i in ${ioarray[@]};do
```

```
eval temp="${!i}"
```

#Replace the special character @, and the space in the shell will be truncated to two elements

```
temp=${temp/@/ }
```

```
echo "exec,${temp}"
```

#Ensure that the final output is in the following format. The first character is the measurement name. If the  
input.exec plug-in has the configuration name "suffix", the suffix will be added automatically

#The output format is measurement name, comma, tag keys (comma separated), space, filed keys (comma  
separated)

#The data format output mismatch will lead to the failure of telegraf to parse the data and go to the influxdb. It  
took a long time to debug and didn't look at the hole dug by the official website

```
#exec,datadir=/data/data11 r=4.1,w=6.1,await=0.83,svctm=1.35,util=1.46"
done
#echo ${devdict[@]}
```

```
[[inputs.exec]]
  ##Commands array
  commands = ["bash /appcom/telegraf/collect_iostat.sh",]
  timeout='5s'
  ##Suffix for measurements
  name_suffix="_collectiostat"
  data_format="influx"
```

## Sample #2

```
#!/bin/sh
hostname=`hostname`
uptime=`awk '{print $1}' /proc/uptime`
if uptime |grep -q user ; then
load1=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $7}'`
load5=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $8}'`
load15=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $9}'`
else
load1=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $5}'`
load5=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $6}'`
load15=`uptime | grep -ohe 'up .*' | sed 's/,//g' | awk '{ print $7}'`
fi
echo "uptime,host=$hostname uptime=$uptime,load1=$load1,load5=$load5,load15=$load15"
```

```
[agent]
interval = "5s"
round_interval = true
[[inputs.swap]]
  [inputs.swap.tags]
    metrics_source="telegraf_demo"
[[inputs.exec]]
  commands = ["/etc/telegraf/uptime.sh"]
  data_format = "influx"
  [inputs.exec.tags]
    metrics_source="telegraf_demo"
[[outputs.influxdb]]
```

```
url = "https://influxdemo:8086"
database = "telegraf"
```

### Sample #3

```
#!/bin/bash
/usr/bin/speedtest --format json | jq '.download.bandwidth = .download.bandwidth / 125000 | .upload.bandwidth
= .upload.bandwidth / 125000'
```

```
[[inputs.exec]]
  commands = [
    "/home/rock64/speedtest.sh"
  ]
  interval = "300s"
  timeout = "60s"
```

### Sample #4

```
[[inputs.exec]]
  commands = ["sh -c 'sysctl -n dev.cpu.0.temperature | tr -d C'"]
  name_override = "cpu_temp"
  timeout = "5s"
  data_format = "value"
  data_type = "float"
[inputs.exec.tags]
  core = "core0"

[[inputs.exec]]
  commands = ["sh -c 'sysctl -n dev.cpu.1.temperature | tr -d C'"]
  name_override = "cpu_temp"
  timeout = "5s"
  data_format = "value"
  data_type = "float"
[inputs.exec.tags]
  core = "core1"

[[inputs.exec]]
  commands = ["sh -c 'sysctl -n dev.cpu.2.temperature | tr -d C'"]
  name_override = "cpu_temp"
  timeout = "5s"
```

```

data_format = "value"
data_type = "float"
[inputs.exec.tags]
  core = "core2"

[[inputs.exec]]
  commands = ["sh -c 'sysctl -n dev.cpu.3.temperature | tr -d C'"]
  name_override = "cpu_temp"
  timeout = "5s"
  data_format = "value"
  data_type = "float"
[inputs.exec.tags]
  core = "core3"

```

## Q & A

### [agent] Error terminating process: operation not permitted

Causation: ? telegraf.conf ?????? agent ?????????? timeout ?????????? agent ??????????telegraf  
 ?????? agent ???

Solution: ?????????????? agent ?????????? timeout ??????????????????????

????????????? timeout ?????? agent ??????????????????????????????????

?? telegraf ??? agent ?????????????????? timeout ???

???????agent ?? sudo ??? db2 ???????????

```

[[inputs.exec]]
  interval = "1h"
  commands = ["sudo -u db2mon sh -c '/home/db2mon/bin/collect_db2x1h.sh -d centdb -a b_centdb'"]
  timeout = "5s"
  data_format = "influx"

```

?? telegraf ?? kill ? sudo ?????????????????????? collect\_db2x1h.sh????? telegraf ?? sudo  
 ??????

```

[[inputs.exec]]
  interval = "1h"
  commands = ["/home/db2mon/bin/collect_db2x1h.sh -d centdb -a b_centdb"]

```

```
timeout = "15s"
data_format = "influx"
```

?????timeout ?????????? agent????????????????

```
## [inputs.exec] Error in plugin: exec: command timed out for command
'/home/db2mon/bin/collect_db2x1h.sh -d centdb -a b_centdb'
```

????????????? timeout?

## Error in plugin: metric parse error: expected tag at 7:20:

Causation: ??? Influxdata ???????

Solution: ??? 7 ??? 20 ?????Influxdada ???

```
measurement, tag-key1=tag-value1,tag-key2=tag-value2 field-key1=field-value1,field-
key2=field-value2,...
```

- tag-key type: string
- tag-value type: string  
NOTE: ????????
- field-key type: string
- field-value type: Float | Integer | UInteger | String | Boolean  
NOTE: ??? string ??????

## max-series-per-database limit exceeded: (1000000)

Causation: ????????????????????? 1000000?

? InfluxDB CLI ??????????????????

```
show series cardinality on <db-name>
```

Solution: ?? InfluxDB ?????????? /etc/influxdb/influxdb.conf ??? 1000000

```
# max-series-per-database = 1000000
max-series-per-database = 2000000
```

?? InfluxDB

```
systemctl restart influxdb
```



---

Revision #43

Created 16 June 2021 06:35:56 by Admin

Updated 2 February 2024 13:19:34 by Admin