Part 41
-
Telegraf

Version: 2020-02-20

Introduction

Telegraf collects metrics and events from systems, databases, IoT sensors and much more and sends these metrics to a database, most likely Telegraf. Telegraf is plugin-driven and has a multitude of plugins already installed.

Installing Telegraf to the Raspberry Pi

The first thing we need do before installing Telegraf to the Raspberry Pi is making sure that all the currently installed packages are up to date.
We can upgrade all installed packages by running the following two commands.

```
sudo apt update
sudo apt upgrade
```

and reboot now.

```
sudo reboot
```

With everything now up to date, we can now proceed with installing Telegraf to the Raspberry Pi. Our next step is to add the Telegraf repository key to our Raspberry Pi.
Adding the key will allow the package manager on Raspbian to search the repository and verify the packages its installing.

We can add the Telegraf key by running the following command.

```
wget -qO- https://repos.influxdata.com/telegraf.key | sudo apt-key add -
```

This command will download the key using wget and pass it directly into the **`apt-key`** program by using a pipe "|".

Now that we have the Telegraf repository key installed, we will need to go ahead and add its repository to the sources list. Enter the following command to do so. Make sure you pick the right command for the version of Raspbian that you are running. Most users running on new installations of Raspbian will likely be running Raspbian Buster.

```
echo "deb https://repos.influxdata.com/debian buster stable" | sudo tee
/etc/apt/sources.list.d/telegraf.list
```

With the repository added, we now need to go ahead and update the package list again.
We need to do this so that the apt package manager searches the repository that we just added for packages. The operating system does not automatically do this. Run the following command on your Raspberry Pi to update the package list.

```
sudo apt update
```

Now that we have set up the repository, we can now move on to installing the Telegraf software.

To install Telegraf to our Raspberry Pi, all we need to do is run the command below.

```
sudo apt -y install telegraf
```

With Telegraf now installed, let's now get it to start at boot. We can do this by making use of the systemctl service manager to enable our Telegraf service file. Run the following two commands to enable Telegraf to start at boot.

```
sudo systemctl unmask telegraf
sudo systemctl enable telegraf
```

The first command we use unmasks the telegraf service file. Unmasking the service ensures that we can enable and start the service as a masked service is unable to be started.

Our second command enables the telegraf service. This command will tell the service manager to keep an eye on the "telegraf.service" file and setup the service based on its contents.
Now that everything has been set up, we can now proceed to start up Telegraf.

To start up the Telegraf server, we will need to run the following command. The service manager will then start up the service and begin monitoring it.

```
sudo systemctl start telegraf
```

Note: The way to install in according the the website of Telegraf is

```
wget https://dl.influxdata.com/telegraf/releases/telegraf-
1.13.4_linux_armhf.tar.gz
tar xf telegraf-1.13.4_linux_armhf.tar.gz
```

However,

- this make is version dependent. So you need to check at installation time for the right version
- You will have to do the installation yourself = putting files to the right place, setting up the daemon, autostart etc.

Location of filesystem

Telegraf runs as a daemon using the new `systemd` way. The daemon config file `telegraf.service` is located in `/usr/lib/telegraf/scripts/`

```
nano telegraf.service

[Unit]
Description=The plugin-driven server agent for reporting metrics into InfluxDB
Documentation=https://github.com/influxdata/telegraf
After=network.target

[Service]
EnvironmentFile=-/etc/default/telegraf
User=telegraf
ExecStart=/usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-
directory /etc/telegraf/telegraf.d $TELEGRAF_OPTS
ExecReload=/bin/kill -HUP $MAINPID
Restart=on-failure
RestartForceExitStatus=SIGPIPE
KillMode=control-group

[Install]
WantedBy=multi-user.target
```

As you can see, it picks up the base config file `telegraf.conf` from `/etc/telegraf/ and will read all other config files from /etc/telegraf/telegraf.d`

Configuring Telegraf

Besides a small demo config below, you can also find more details on the config file in my other document `Windows – Telegraf.pdf` where I provide a config file to monitor your Pi's using SNMP.

I will use the script of Rob Cowart which you can find on Github
`https://github.com/robcowart/raspberry_pi_stats.`

We modify it a bit to output to file instead of an influx database but if you want, you can have the output going to influxDB as well.

Make sure you are in home directory

```
cd /home/pi
git clone https://github.com/robcowart/raspberry_pi_stats
cd raspberry_pi_stats
nano telegraf.conf
```

change

```
commands = [ "/usr/local/bin/rpi-stats.sh" ]
```

to

```
commands = [ "/home/pi/raspberry_pi_stats/rpi-stats.sh" ]
```

comment out all of the `outputs.influxd`b section.

Add

```
[[outputs.file]]
  ## Files to write to, "stdout" is a specially handled file.
  files = ["/home/pi/telegraf.out"]
```

Save and exit

copy file to /etc/telegraf

```
cp telegraf.conf /etc/telegraf/telegraf.conf
```

make the rpi-stats.sh executable

```
chmod 777  rpi-stats.sh
```

run it to test it

```
./rpi-stats.sh
```

you should see something like this

```
raspberry_pi,host=MeRasPi4B-Test
soc_temp=41.0,arm_freq=600169920i,core_freq=199995120i,h264_freq=0i,isp_freq=0i,
v3d_freq=360558112i,uart_freq=48001464i,pwm_freq=0i,emmc_freq=250000496i,pixel_f
req=75001464i,vec_freq=0i,hdmi_freq=0i,dpi_freq=0i,core_volts=0.8563,sdram_c_vol
ts=1.1,sdram_i_volts=1.1,sdram_p_volts=1.1,config_arm_freq=1500000000i,config_co
```

```
re_freq=500000000i,config_gpu_freq=500000000i,config_sdram_freq=0i,arm_mem=99800
0000i,gpu_mem=16000000i,malloc_total_mem=4000000i,malloc_mem=4000000i,reloc_tota
l_mem=10000000i,reloc_mem=9000000i,oom_count=0i,oom_ms=0i,mem_reloc_allocation_f
ailures=0i,mem_reloc_compactions=0i,mem_reloc_legacy_block_failures=0i
```

First run a test with telegraf

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

you should see the same

Running this with the daemon won't work as there is a problem using stdout when run as a daemon. So run in as follows

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

wait a minute and hit Crtl-C to stop it. Now you can check the file telegraf.out.

So that's working. If you want to run telegraf on your Pi to check out all your Pi's, you can copy the `rpi.conf` file that you can find in my `Windows - Telegraf.pdf` document to your Pi with the name `telegraf.conf` into `/etc/telegraf` and run it with the telegraf daemon.

First make sure it works

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

Next, restart the telegraf daemon

```
sudo systemctl restart telegraf
```