

Building Apps With Time Series Data the Easy Way

Agenda

Why app, cloud developers increasingly use time series data

Time series for app performance, cloud-native projects and more

InfluxData's Telegraf collects and publishes metrics, monitoring data, events

Bio

Advocate for InfluxData's Telegraf open source server agent for metrics

Long-time active contributor to open source projects

Prior to InfluxData, worked on Ubuntu



Alan Pope

Developer Advocate



influxdata



Building Apps With Time Series Data The Easy Way

Alan Pope, Developer Advocate

InfluxData

Today's Presentation

Telegraf

- * Introduction to Telegraf
- * Telegraf plugins
- * Gathering metrics using Telegraf
- * Beyond existing plugins
- * Data transformation

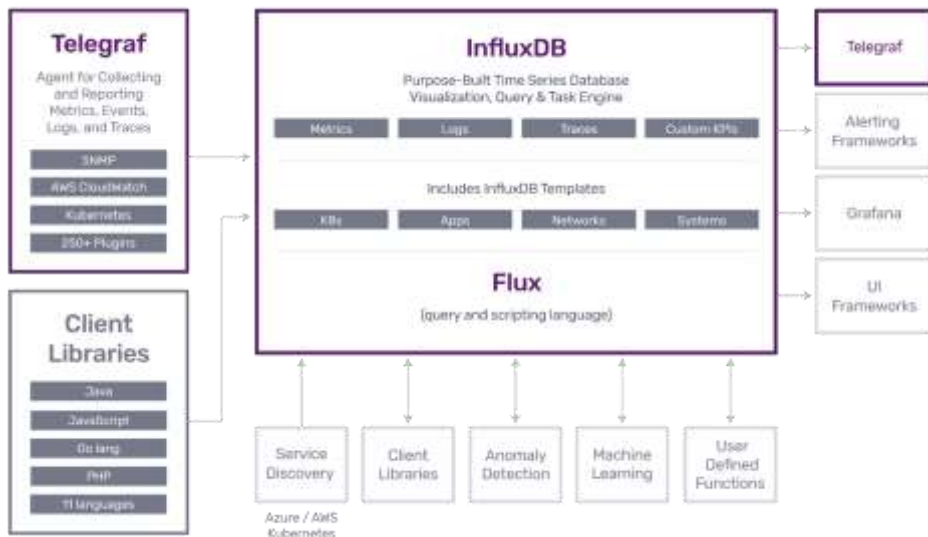




Telegraf Agent: Core functionality

Introduction to Telegraf

- Active Community
- Over 200 plugins
 - 170+ Input plugins
 - 50+ Output plugins
 - 20+ Processor plugins
 - 5+ Aggregator plugins
 - 5+ Serializers
- Single Binary
- Extendable in any language
- Low and no code options



Telegraf Agent

- Core of telegraf functionality providing the following
 - Robust scheduler
 - Adjustments for clock-drift
 - Adjustments for job scheduling issues that may occur
 - In-memory metric buffers
 - Metric tracking with flow back-pressure in plugins like Kafka
 - Full-streaming support
 - Metric routing: name & field pass & drop
 - Flexible parsing, formatting, serializing





Telegraf Plugins

Telegraf Plugin Types

170+

Input

50+

Output

20+

Processors

5+

Aggregators



Input Plugins

activemq
aerospike
amqp_consumer
apache
apcupsd
aurora
azure_storage_queue
bcache
beanstalkd
bind
bond
burrow
cassandra
ceph
cgroup

chrony
cisco_telemetry_mdt
clickhouse
cloud_pubsub
cloud_pubsub_push
cloudwatch
conntrack
consul
couchbase
couchdb
cpu
dcos
disk
diskio
disque

dmcache
dns_query
docker
docker_log
dovecot
ecs
elasticsearch
ethtool
eventhub_consumer
exec
execd
fail2ban
fibaro
file
filecount

filestat
fireboard
fluentd
github
gnmi
graylog
haproxy
hddtemp
http
http_listener_v2
http_response
httpjson
icinga2
infiniband
influxdb



Input Plugins

influxdb_listener
influxdb_v2_listener
intel_rdt
internal
interrupts
ipmi_sensor
ipset
iptables
ipvs
jenkins
jolokia
jolokia2
jti_openconfig_telemetry
kafka_consumer
kafka_consumer_legacy

kapacitor
kernel
kernel_vmstat
kibana
kinesis_consumer
kube_inventory
kubernetes
lanz
leofs
linux_sysctl_fs
logparser
logstash
lustre2
mailchimp
marklogic

mcrouter
mem
memcached
mesos
minecraft
modbus
mongodb
monit
mqtt_consumer
multifile
mysql
nats
nats_consumer
neptune_apex
net

net_response
nginx
nginx_plus
nginx_plus_api
nginx_sts
nginx_upstream_check
nginx_vts
nsd
nsq
nsq_consumer
nstat
ntpq
nvidia_smi
opcua
openldap



Input Plugins

openntpd
opensmtpd
openweathermap
passenger
pf
pgbouncer
phpfpm
ping
postfix
postgresql
postgresql_extensible
powerdns
powerdns_recursor
processes
procstat

prometheus
proxmox
puppetagent
rabbitmq
raindrops
ras
redfish
redis
rethinkdb
riak
salesforce
sensors
sflow
smart
snmp

snmp_legacy
snmp_trap
socket_listener
solr
sqlserver
stackdriver
statsd
suricata
swap
synproxy
syslog
sysstat
system
systemd_units
tail

tcp_listener
teamspeak
temp
tengine
tomcat
trig
twemproxy
udp_listener
unbound
uwsgi
varnish
vsphere
webhooks
win_eventlog
win_perf_counters

win_services
wireguard
wireless
x509_cert
zfs
zipkin
zookeeper



Processor Plugins

clone
converter
date
dedup
defaults
enum
execd
filepath
ifname
override
parser
pivot

port_name
printer
regex
rename
reverse_dns
s2geo
starlark
strings
tag_limit
template
topk
unpivot



Aggregator Plugins

basicstats
final
histogram
merge
minmax
valuecounter



Output Plugins

amon
amqp
application_insights
azure_monitor
cloud_pubsub
cloudwatch
cratedb
datadog
discard
dynatrace
elasticsearch
exec
execd
file
graphite

graylog
health
http
influxdb
influxdb_v2
instrumental
kafka
kinesis
librato
logzio
mqtt
nats
newrelic
nsq
opentsdb

prometheus_client
riemann
riemann_legacy
socket_writer
stackdriver
sumologic
syslog
timestream
warp10
wavefront
yandex_cloud_monitoring



Influxdb cloud

Create your Free InfluxDB Cloud Account

No credit card required

Continue with

Google

Microsoft

OR

Log In Sign Up

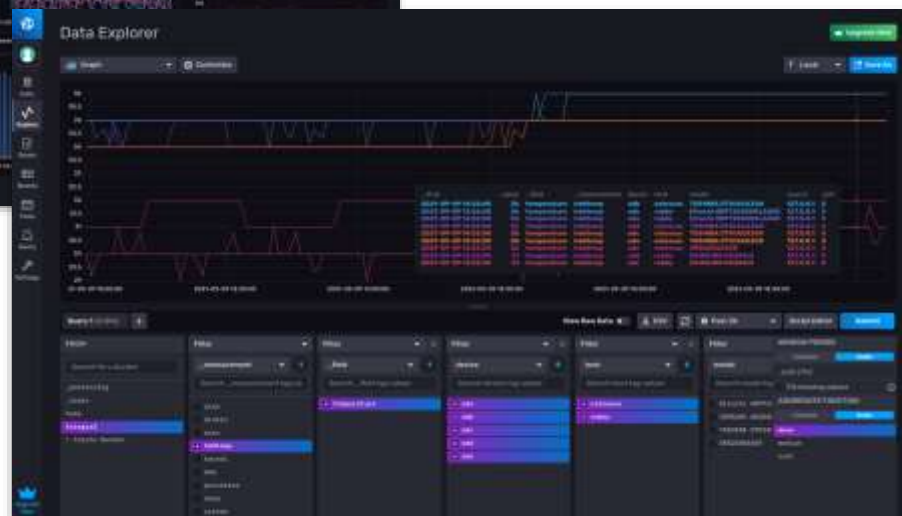
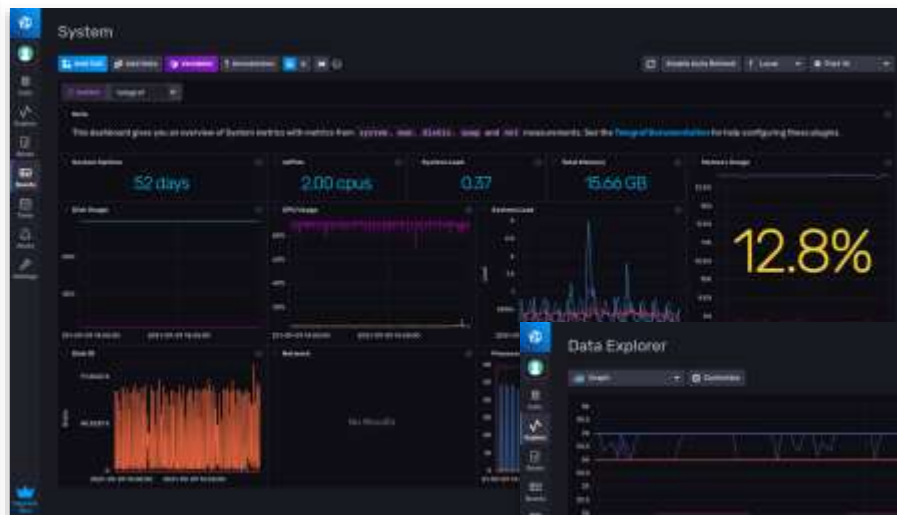
First Name* Last Name*

Work Email Address*

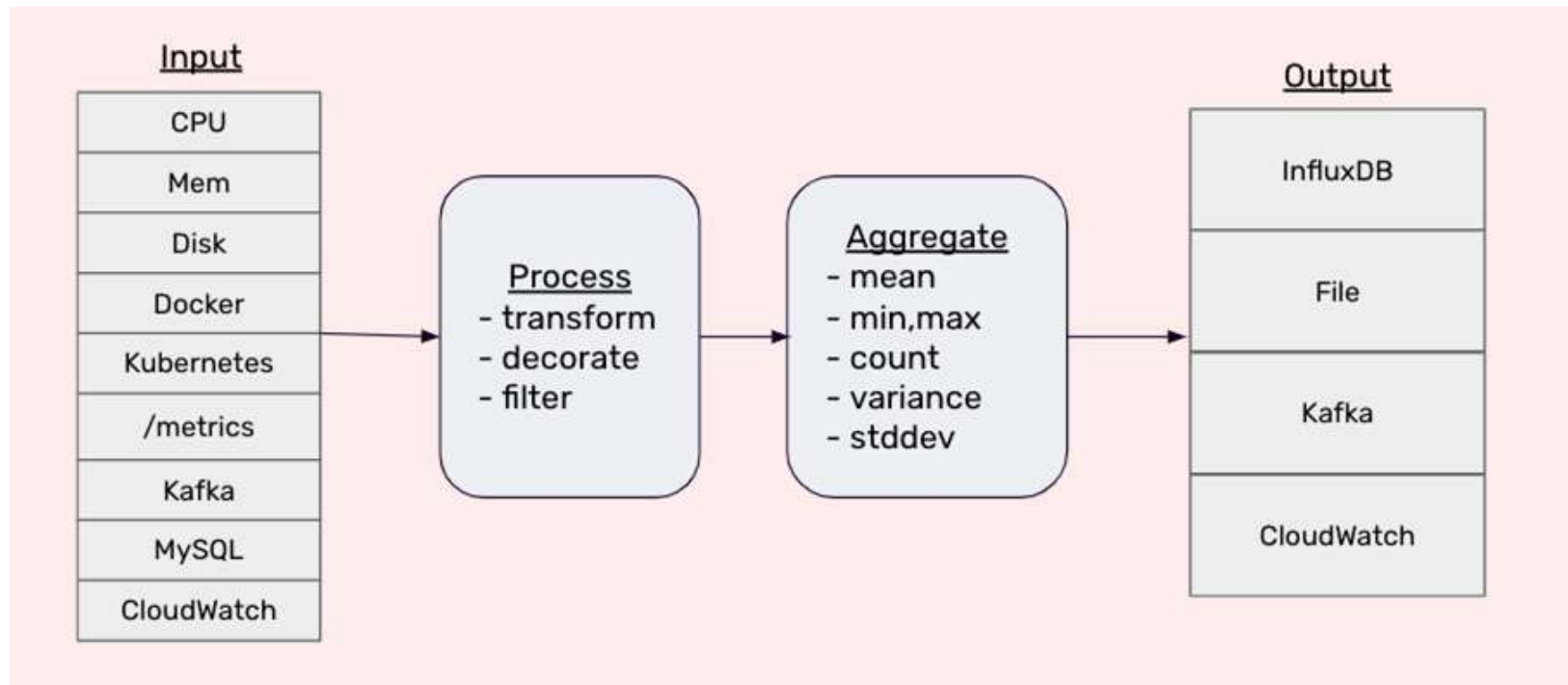
Password*

Confirm Password*

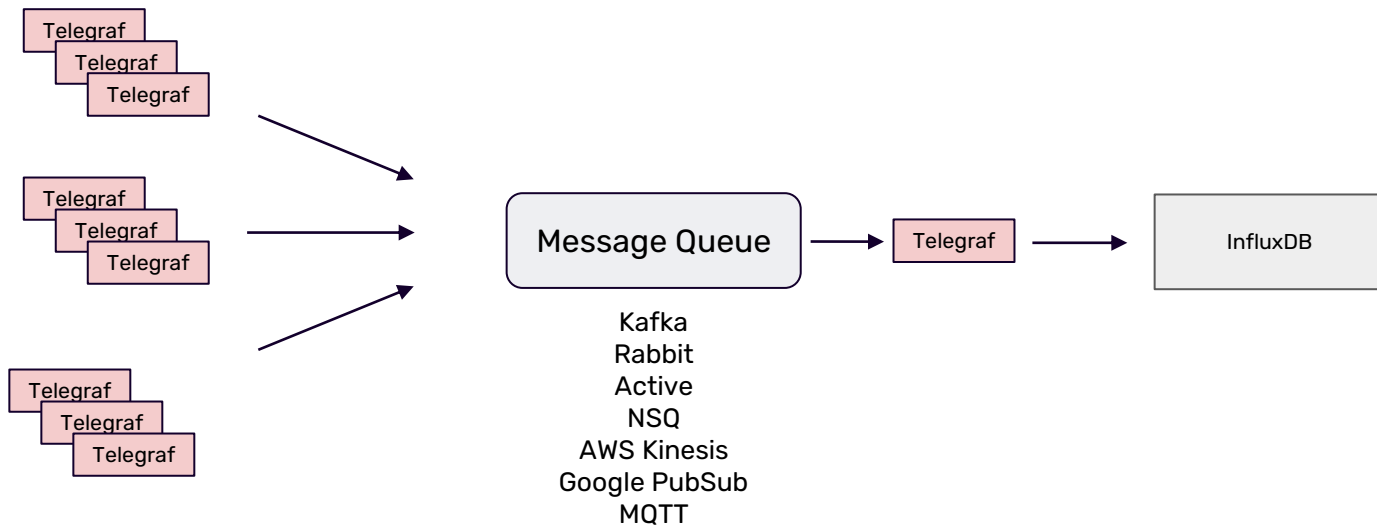
Create Account



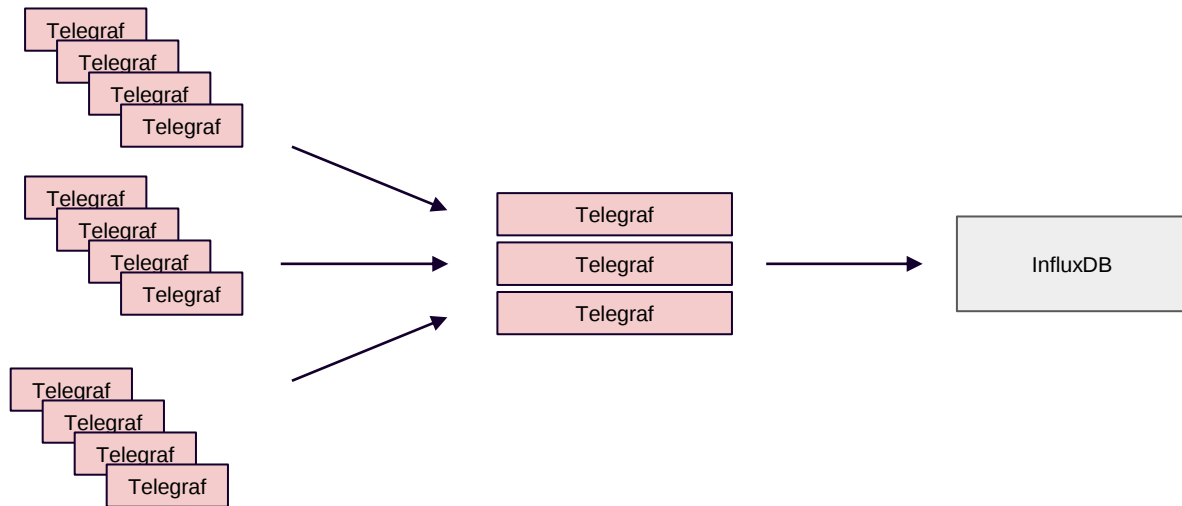
One Telegraf, Multiple Plugins



Telegraf - Best Practice



Telegraf





Gathering metrics using Telegraf

What's in a configuration file?

The Telegraf config file needs to be specified for Telegraf agent to operate properly. It contains setup for the agent, global tags, and enabled outputs (through commenting out or removing unnecessary lines)

```
# Global tags can be specified here in key="value" format.
[global_tags]
  # dc = "us-east-1" # will tag all metrics with dc=us-east-1
  # rack = "1a"
  ## Environment variables can be used as tags, and throughout the config file
  # user = "$USER"

# Configuration for telegraf agent
[agent]
  ## Default data collection interval for all inputs
  interval = "10s"
  ## Rounds collection interval to 'interval'
  ## ie, if interval="10s" then always collect on :00, :10, :20, etc.
  round_interval = true
```

```
[global_tags]
[agent]
  interval = "2s"
  flush_interval = "2s"
  debug = true
[[inputs.cpu]]
[[outputs.file]]
```



Set and Test Your Configuration

WRITE DESIRED CONFIG FILE

```
$ telegraf --input-filter cpu:system --output-filter influxdb --processor-filter starlark --aggregator-filter valuecounter config > telegraf.conf
```

RUN CONFIG FILE TEST

```
$ telegraf --config telegraf.conf --test
```

VERIFY YOUR CONFIG WORKS

```
Jessicas-MacBook-Pro:my-telegraf-fork jess_ingrassellino$ telegraf --config telegraf.conf --test
2020-11-18T00:20:52Z I! Starting Telegraf 1.15.3
> system,host=Jessicas-MBP load1=1.57470703125,load15=1.45947265625,load5=1.5048828125,n_cpus=12i,n_users=2i 1605658852000000000
> system,host=Jessicas-MBP uptime=171657i 1605658852000000000
> system,host=Jessicas-MBP uptime_format="1 day, 23:40" 1605658852000000000
> cpu,cpu=cpu0,host=Jessicas-MBP usage_guest=0,usage_guest_nice=0,usage_idle=98.00000000098953,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softi
rq=0,usage_steal=0,usage_system=0,usage_user=2.0000000001018634 1605658853000000000
> cpu,cpu=cpu1,host=Jessicas-MBP usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_ste
al=0,usage_system=0,usage_user=0 1605658853000000000
> cpu,cpu=cpu2,host=Jessicas-MBP usage_guest=0,usage_guest_nice=0,usage_idle=96.07843137182171,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softi
rq=0,usage_steal=0,usage_system=1.9607843136433174,usage_user=1.9607843137324836 1605658853000000000
> cpu,cpu=cpu3,host=Jessicas-MBP usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_ste
```



Plugin Documentation

The screenshot shows the InfluxData documentation website for Telegraph v1.19. The page is titled "Telegraf plugins" and provides an overview of the plugin-driven architecture. It includes a sidebar with navigation links for guides, introduction, concepts, and specific plugin types. The main content area features a description of Telegraph and a series of filterable lists for plugin types, categories, and operating systems. A "Jump to:" section at the bottom provides direct links to input, output, aggregator, and processor plugins.

Telegraf v1.19 | Documentation

Search Telegraph v1.19

- **Telegraf v1.19**
- Guides
- Introduction
- Concepts
- **Telegraf plugins**
- Data formats
- Administration
- About the project

INFLUXDATA PLATFORM

- Introduction
- Get started
- Install and deploy
- Monitor
- Integrate
- Troubleshoot
- Frequently asked questions

Telegraf plugins

Telegraf is a plugin-driven agent that collects, processes, aggregates, and writes metrics. It supports four categories of plugins including input, output, aggregator, and processor.

Plugin type	Plugin category	
☐ Input (234)	☐ Applications (29)	☐ Logging (102)
☐ Output (449)	☐ Build & Deploy (18)	☐ Messaging (23)
☐ Aggregator (10)	☐ Cloud (28)	☐ Networking (42)
☐ Processor (26)	☐ Containers (10)	☐ Servers (29)
☐ External (11)	☐ Data Stores (33)	☐ Systems (30)
	☐ IoT (10)	☐ Web (31)

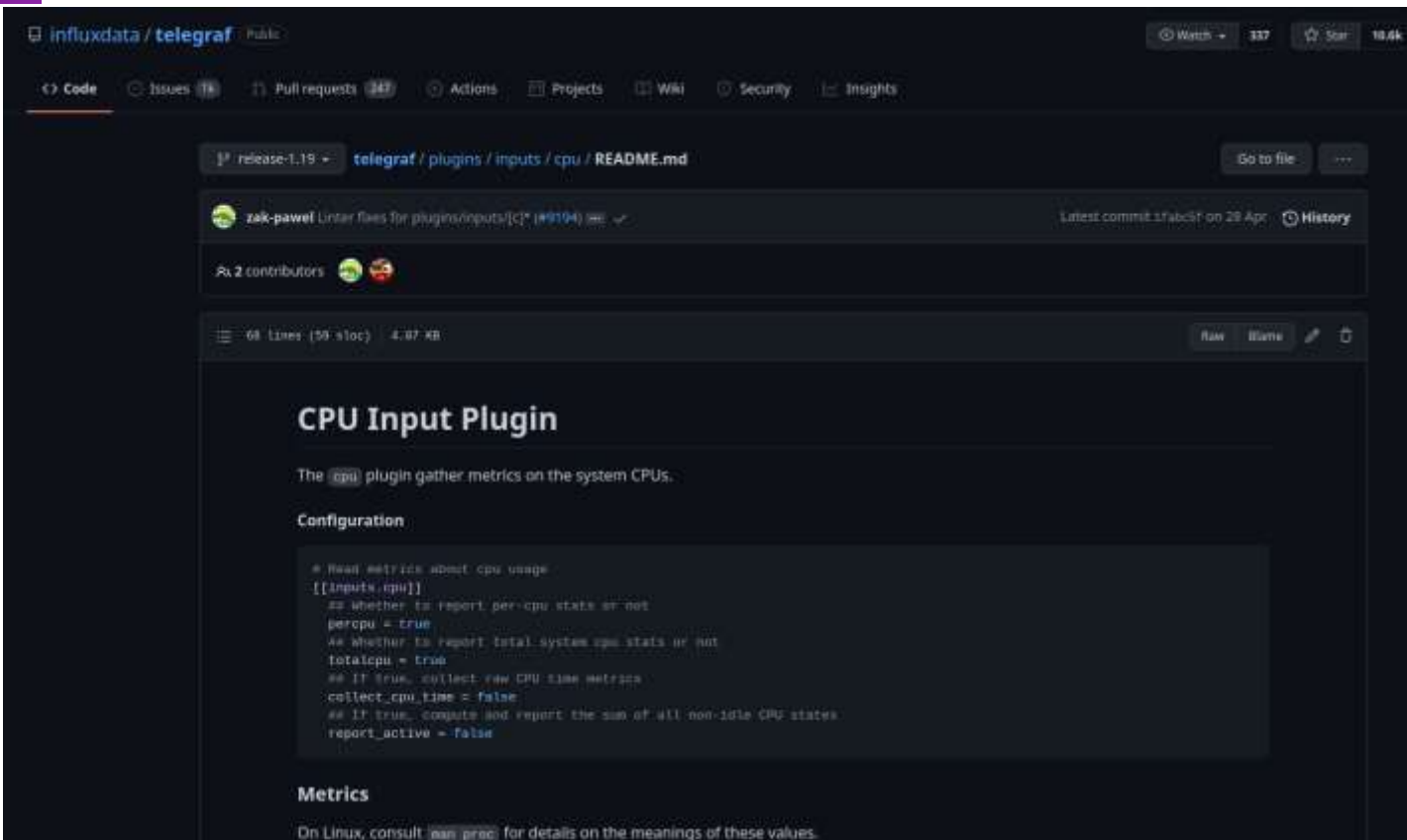
Operating system	Status
☐ Linux (270)	☐ New (10)
☐ macOS (274)	☐ Deprecated (0)
☐ Windows (228)	

Jump to:

- **Input plugins**
- **Output plugins**
- **Aggregator plugins**
- **Processor plugins**



Plugin Documentation



The screenshot shows the GitHub interface for the `influxdata/telegraf` repository. The file path is `telegraf / plugins / inputs / cpu / README.md`. The commit is by `zak-pawel` with the message "Lint fixes for plugins/inputs/cpu" (#9104). It shows 2 contributors and 68 lines of code. The main heading is "CPU Input Plugin". The description states: "The `cpu` plugin gather metrics on the system CPUs." The "Configuration" section contains a TOML snippet:

```
[[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
```

 The "Metrics" section begins with "On Linux, consult `man proc` for details on the meanings of these values."

influxdata / telegraf Public

Watch 337 Star 10.6k

Code Issues 14 Pull requests 347 Actions Projects Wiki Security Insights

release-1.19 telegraf / plugins / inputs / cpu / README.md Go to file

zak-pawel Lint fixes for plugins/inputs/cpu (#9104) Latest commit 1fab51f on 29 Apr History

2 contributors

68 lines (50 sloc) 4.07 KB Raw Blame

CPU Input Plugin

The `cpu` plugin gather metrics on the system CPUs.

Configuration

```
[[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
```

Metrics

On Linux, consult `man proc` for details on the meanings of these values.





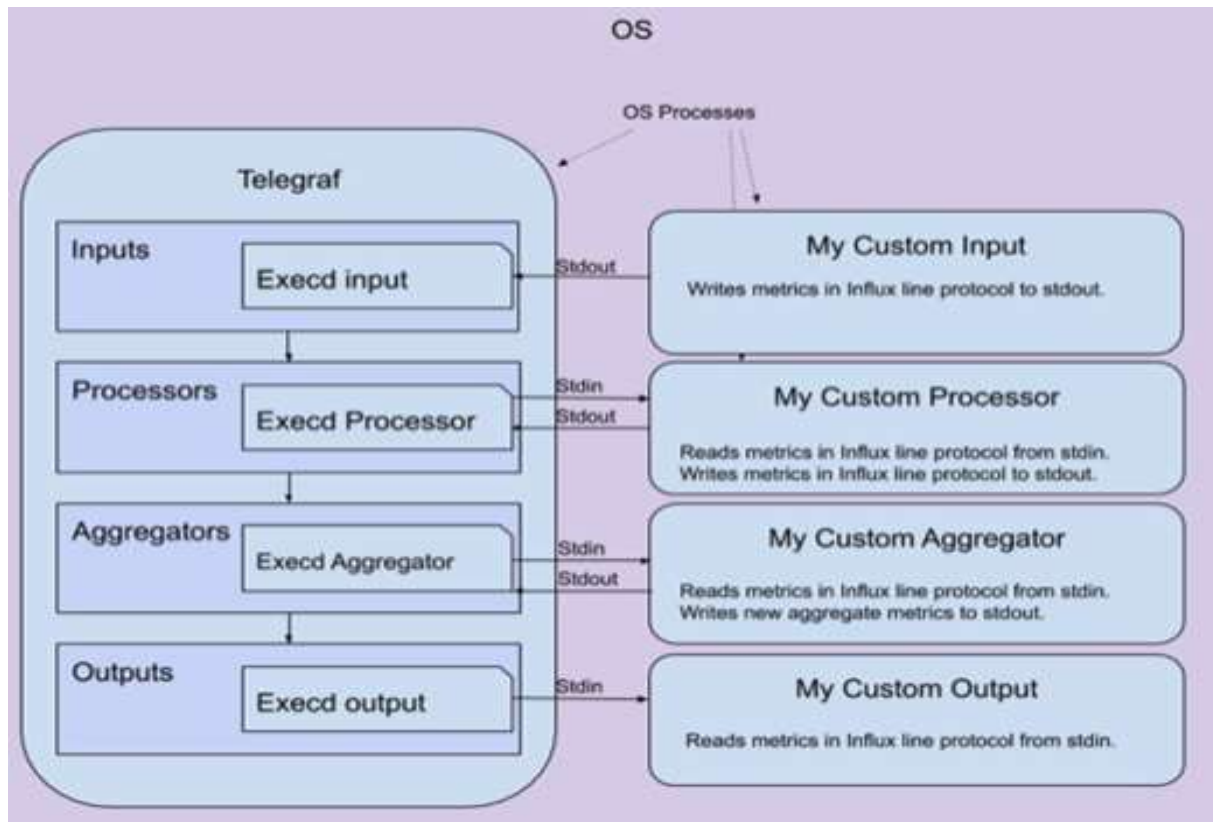
ExecD Plugins

External plugins via ExecD

- Plugin runs in its own process
- Requires line protocol
- Avoid the need for review by Telegraf team
- Supports the same API as an internal plugin
- Can use for non-GO plugins
- Can use for licensed software plugins
- Can use for any type of plugin (input, output, processor, aggregator)



ExecD Plugin Architecture



Sample ExecD Config: Input, Output, Processor

```
[[inputs.execd]]
  command = ["ruby", "plugins/inputs/execd/examples/count.rb"]

[[outputs.execd]]
  command = ["sh", "plugins/outputs/execd/examples/file/file.sh"]
  data_format = "json"

[[processors.execd]]
  command = ["python", "./processors/forecasting.py"]
```



Starlark Processor Plugin

Why Starlark?

- Dialect of Python (formerly Skylark) intended for use as configuration language
- Independent Starlark threads execute in parallel
- Starlark Processor calls a Starlark function for each matched metric, allowing for custom programmatic metric processing.
 - Python code is unlikely to work unmodified
 - execution environment is sandboxed and cannot access file system, network, system resources.
- Starlark processor has the following capabilities:
 - Math and String operations
 - Renaming tags
 - Logic operations



Starlark: Run as embedded function

```
[[processors.starlark]]
# Reads the Starlark script embedded
source = '''
def apply(metric):
    # Get the field called current and put into variable I
    I = metric.fields['current']
    # Get the field called voltage and put into variable V
    V = metric.fields['voltage']
    # Create a new field, power, which is I times V
    metric.fields['power'] = I * V
'''
```



Starlark: Call a starlark file

```
[[processors.starlark]]  
# File containing a Starlark script.  
script = "/usr/local/bin/myscript.star"
```





Summary

Thank You



Q&A

Getting Started with Telegraf

- Telegraf Resources Page <https://bit.ly/3k4xgXn>
- Telegraf Documentation <https://bit.ly/3nk6EDq>
- Telegraf Plugins <https://bit.ly/3laGXCL>
- Telegraf on GitHub <https://github.com/influxdata/telegraf>





Time Series Platform for Database Monitoring & Analytics

<https://www.influxdata.com/>

Overview

Tour InfluxData Offerings

https://www.influxdata.com/get-influxdb/?utm_campaign=whytimeseries&utm_medium=partner&utm_source=idev

- InfluxDB Cloud
- InfluxDB Enterprise
- Open Source

InfluxData Resource Center

<https://www.influxdata.com/resources/>

- Case Studies
- Datasheets
- Documentation
- Technical papers
- Videos
- Webinars

Popular Resources

Considerations When Evaluating Time Series Databases

https://www.influxdata.com/resources/lets-compare-a-benchmark-review-of-influxdb-and-cassandra-3-11-1/?utm_source=vendor&utm_medium=referral&utm_campaign=2021-09-23_spsnr-webinar_application-architecture-summit_idev

InfluxDB Tools and Integrations

<https://docs.influxdata.com/influxdb/v2.0/tools/>

APM Using a Time Series Platform

<https://www.influxdata.com/resources/application-performance-monitoring-apm-using-the-influxdb-time-series-platform/>

Start Monitoring Your Sensors with This InfluxDB Template:

<https://bit.ly/3qn0YHW>

Free Trial & More

Telegraf Portal

<https://bit.ly/3k4xgXn>

Try InfluxDB Cloud

(InfluxDB as a Managed Service)

https://www.influxdata.com/products/influxdb-cloud/?utm_source=vendor&utm_medium=referral&utm_campaign=2021-09-23_spsnr-webinar_application-architecture-summit_idev

Getting Started: Writing Data To InfluxDB

https://www.influxdata.com/blog/getting-started-writing-data-to-influxdb/?utm_source=vendor&utm_medium=referral&utm_campaign=2021-08-26_spsnr-webinar_enterprise-integration-summit_idev