



# The Telegraf Toolbelt

---

It can do that, really?



# David McKay

Developer Advocate

at [InfluxData](#)

[@rawcode](#)

 Scottish

 Has 9 Pets

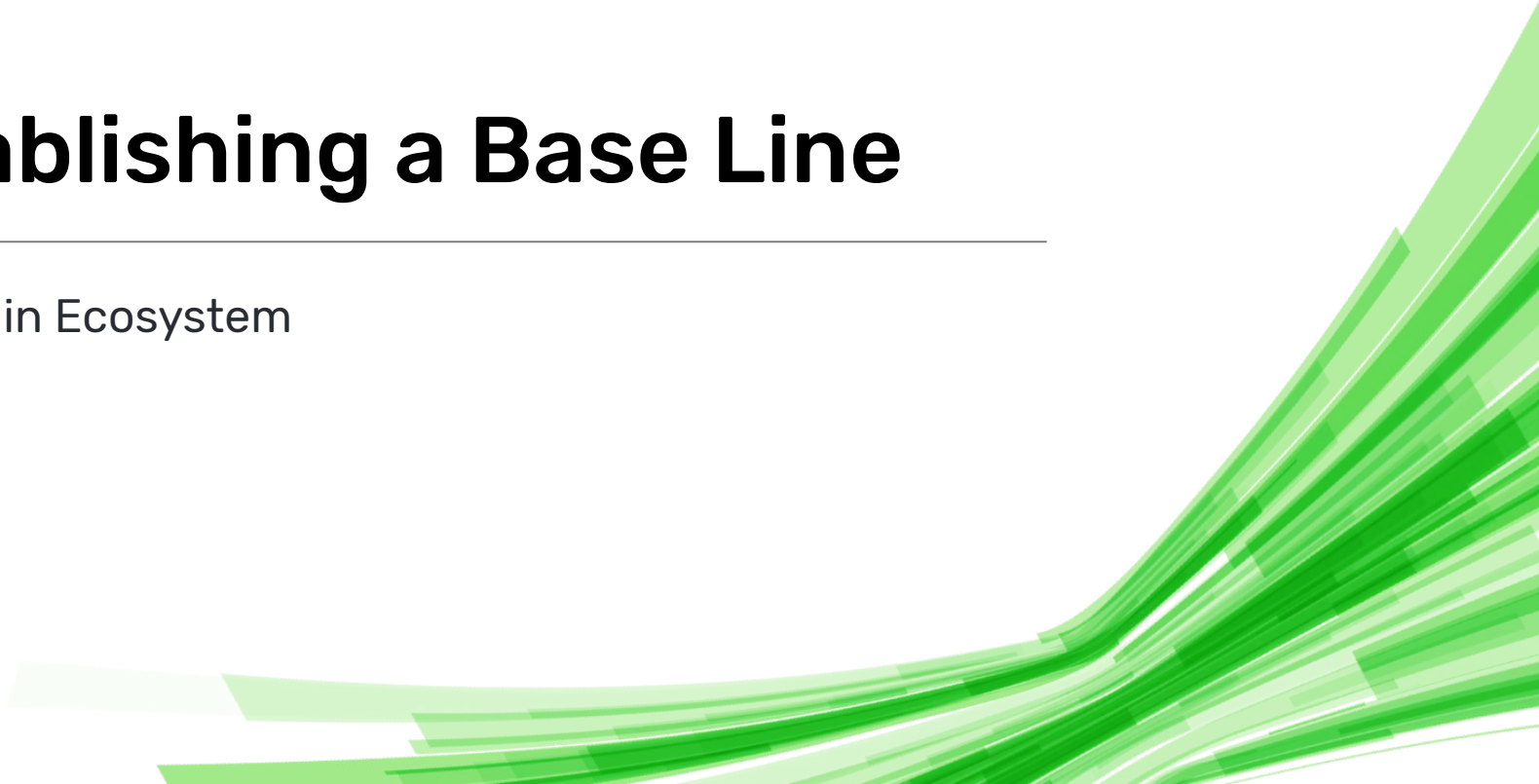
 Esoteric Programming Languages

 Kubernetes Release Team

# Establishing a Base Line

---

The Plugin Ecosystem



# Over 200 Plugins

- 169 Inputs
- 35 Outputs
- 15 Processors
- 14 Parsers
- 9 Serializers
- 8 Aggregators

# V1.10 (May)

## → Inputs

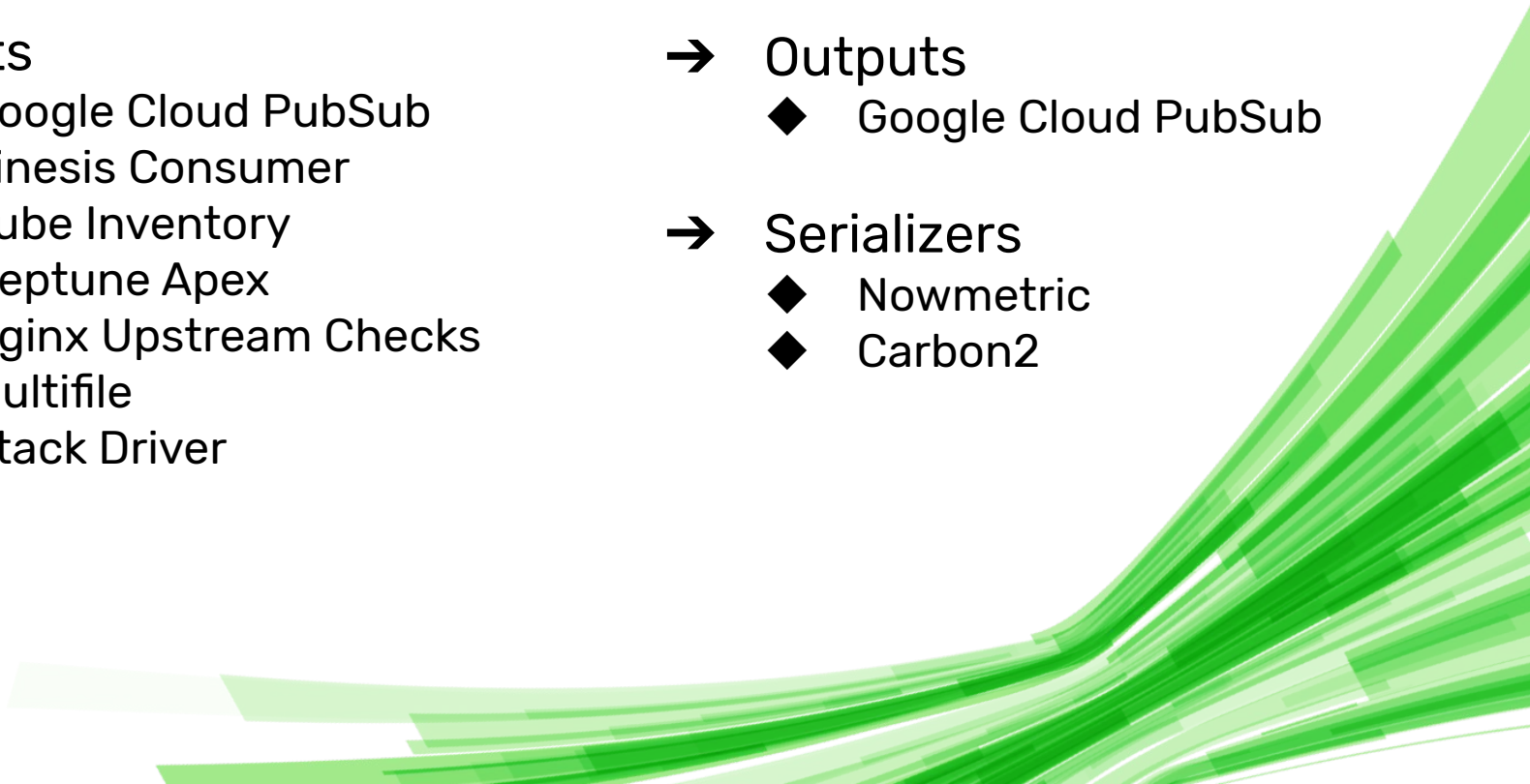
- ◆ Google Cloud PubSub
- ◆ Kinesis Consumer
- ◆ Kube Inventory
- ◆ Neptune Apex
- ◆ Nginx Upstream Checks
- ◆ Multifile
- ◆ Stack Driver

## → Outputs

- ◆ Google Cloud PubSub

## → Serializers

- ◆ Nowmetric
- ◆ Carbon2



# V1.11 (June)

## → Inputs

- ◆ bind9
- ◆ Cisco GNMI
- ◆ Cisco MDT
- ◆ ECS & Fargate
- ◆ GitHub
- ◆ OpenWeatherMap
- ◆ PowerDNS

## → Outputs

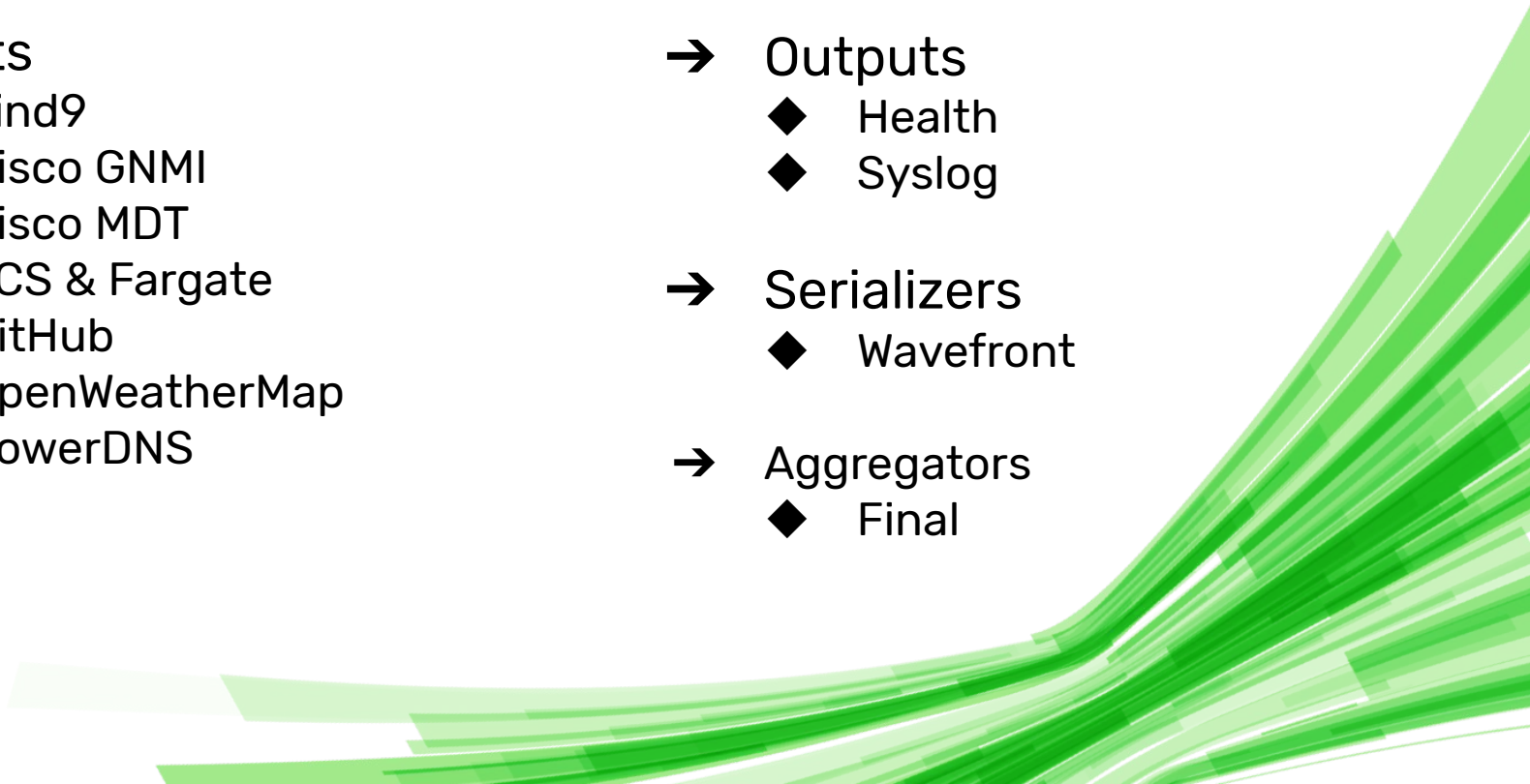
- ◆ Health
- ◆ Syslog

## → Serializers

- ◆ Wavefront

## → Aggregators

- ◆ Final



# V1.12 (September)

## → Inputs

- ◆ apcupsd
- ◆ Docker Logs
- ◆ Fireboard
- ◆ Logstash
- ◆ MarkLogic
- ◆ OpenNTPD
- ◆ uWSGI

## → Outputs

- ◆ Exec

## → Parsers

- ◆ Form

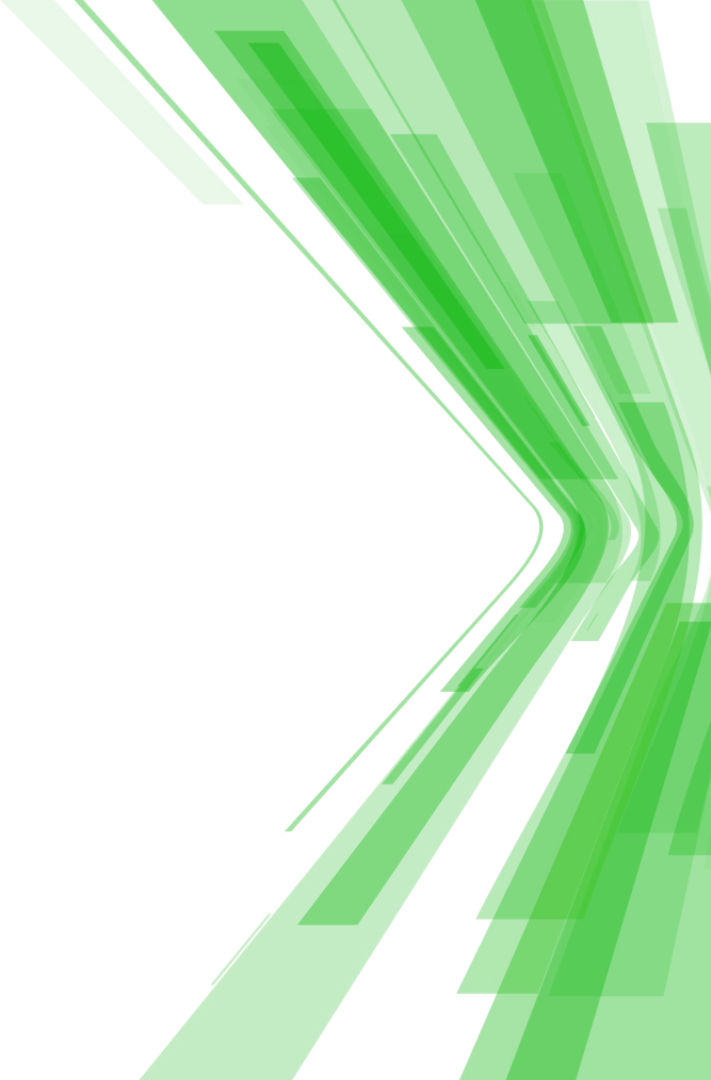
## → Processors

- ◆ Date
- ◆ Pivot
- ◆ Unpivot
- ◆ Tag Limit



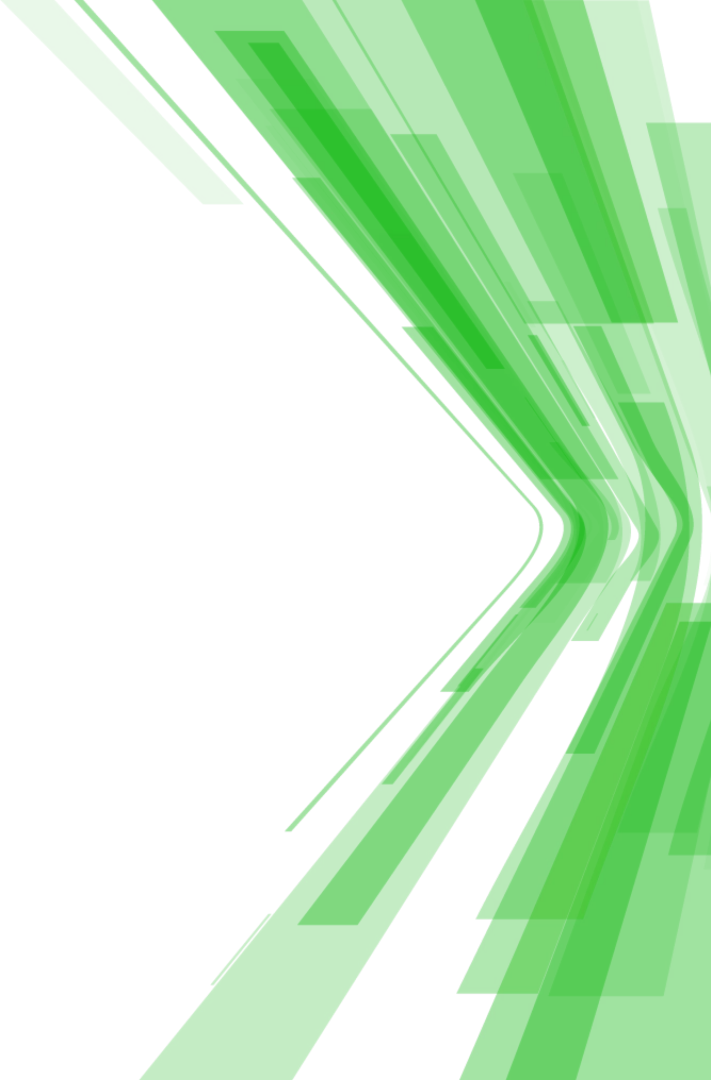


There are **3249**  
**telegraf.conf** files on  
**GitHub**

An abstract graphic on the right side of the slide consisting of multiple overlapping, semi-transparent green lines and shapes that converge towards the right edge, creating a sense of depth and movement.



I could only get **1000**  
**telegraf.conf** files from  
**GitHub**

An abstract graphic on the right side of the slide, consisting of multiple overlapping, semi-transparent green lines and shapes that converge towards the right edge, creating a sense of depth and movement.

I used a sample of 1000  
telegraf.conf files from  
GitHub



# Interval

73% Use 10s (Default)

5.6% use 1s

4% use 5s

2% use 1m

1% use 30s

# Round Interval

---

90% True

# Jitter

---

90% None

Omit  
Hostname

90% False



# Output Plugins

- 71% 1 Output
- 5% 2 Outputs
- 2% 0 Outputs
- .6% 3 Outputs

# Output Plugins

- 72% InfluxDB
- 5% File
- 2% Prometheus Client
- .9% Graphite
- .6% Kafka

# Input Plugins

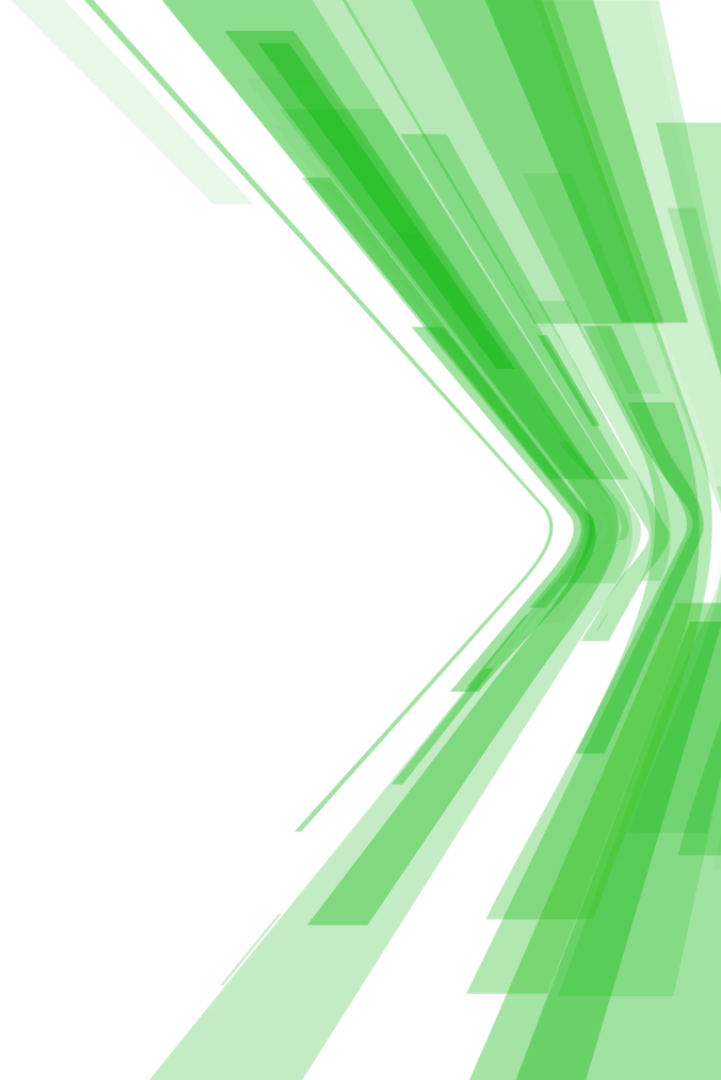
|   |     |    |        |      |
|---|-----|----|--------|------|
| → | 17% | 1  | Input  |      |
| → | 12% | 9  | Inputs |      |
| → | 10% | 8  | Inputs |      |
| → | 5%  | 10 | Inputs |      |
| → | 5%  | 11 | Inputs |      |
| → | 5%  | 6  | Inputs |      |
| → | 5%  | 7  | Inputs |      |
| → | 1   | 56 | Inputs | ?!?! |

# Input Plugins

→ 58% CPU  
→ 53% Mem  
→ 52% Disk  
→ 51% System  
→ 47% DiskIO

→ 47% Swap  
→ 40% Process  
→ 31% Kernel  
→ 28% Docker  
→ 23% Net

**12% Invalid**



# Multiple Outputs

---

Dual Write to InfluxDB 1 and 2

5.6%

A decorative graphic on the right side of the slide, consisting of numerous overlapping, semi-transparent green lines and shapes that create a sense of motion and depth, resembling a stylized wave or a fan of lines.

# Multiple Outputs

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

```
  urls = ["http://influxdb:8086"]
```

```
[[outputs.influxdb_v2]]
```

```
  urls = ["http://influxdb2:9999"]
```

# Remote Configuration

---

Fetching Configuration at Runtime

v1.10

A decorative graphic on the right side of the slide, consisting of numerous overlapping, semi-transparent green lines and shapes that create a sense of motion and depth, resembling a stylized wave or a cluster of data paths.



# Remote Configuration

```
telegraf --config <http_uri>
```

# Remote Configuration

```
telegraf --config
```

```
https://raw.githubusercontent.com  
/influxdata/telegraf/master/etc/telegraf.conf
```

# Remote Configuration

```
SOME_VAR=abc123 telegraf --config <http_uri>
```

# Remote Configuration

```
[agent]  
    interval = "${INTERVAL}"
```

```
[[outputs.influxdb_v2]]  
    token = "${TOKEN}"
```

# Output Resiliency

---

Handling failures without losing metrics

v0.10 (2015!)

A decorative graphic on the right side of the slide, consisting of numerous overlapping, semi-transparent green lines and shapes that fan out from the bottom right towards the top right, creating a sense of motion and depth.

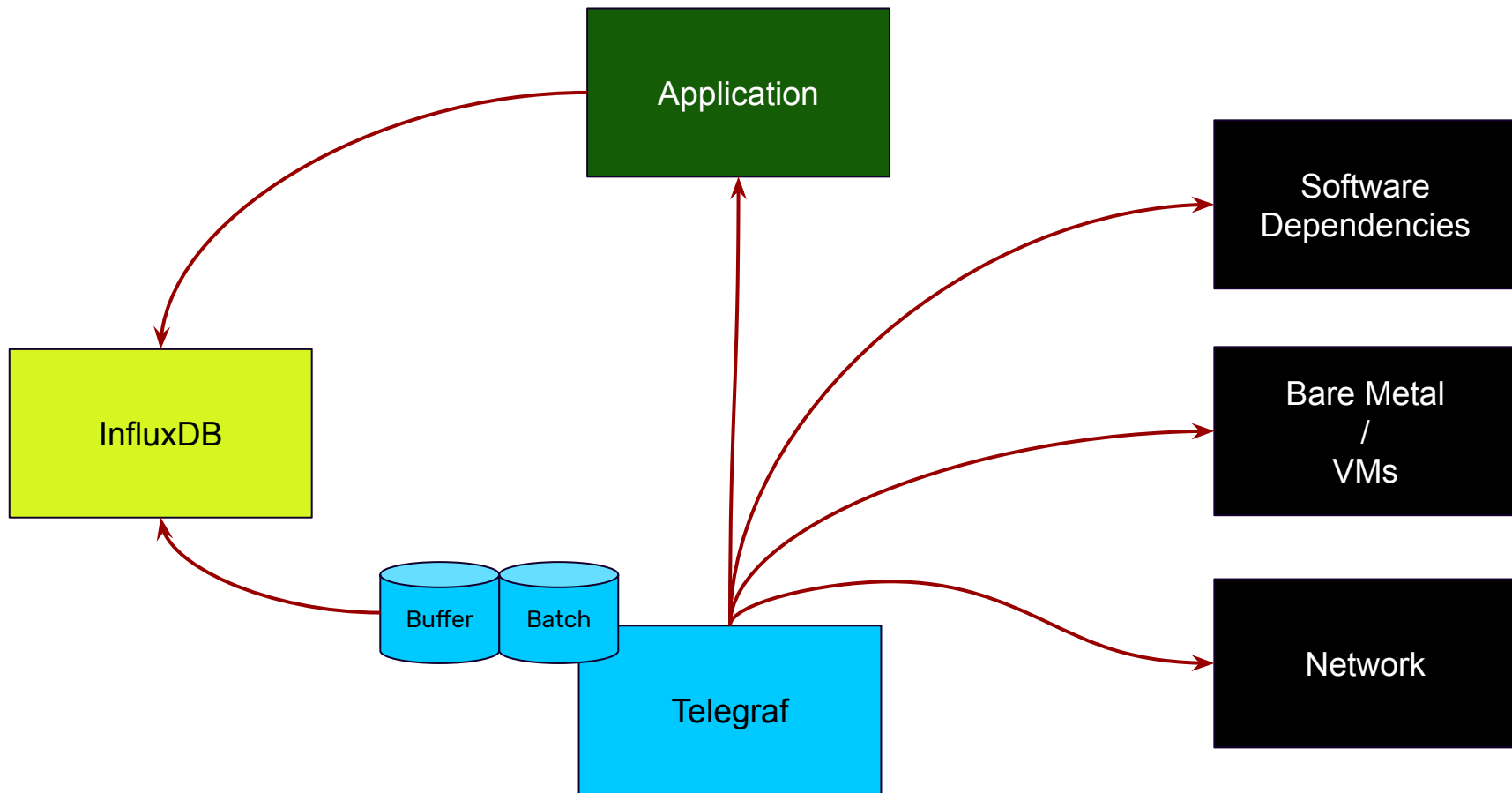
# Output Resiliency

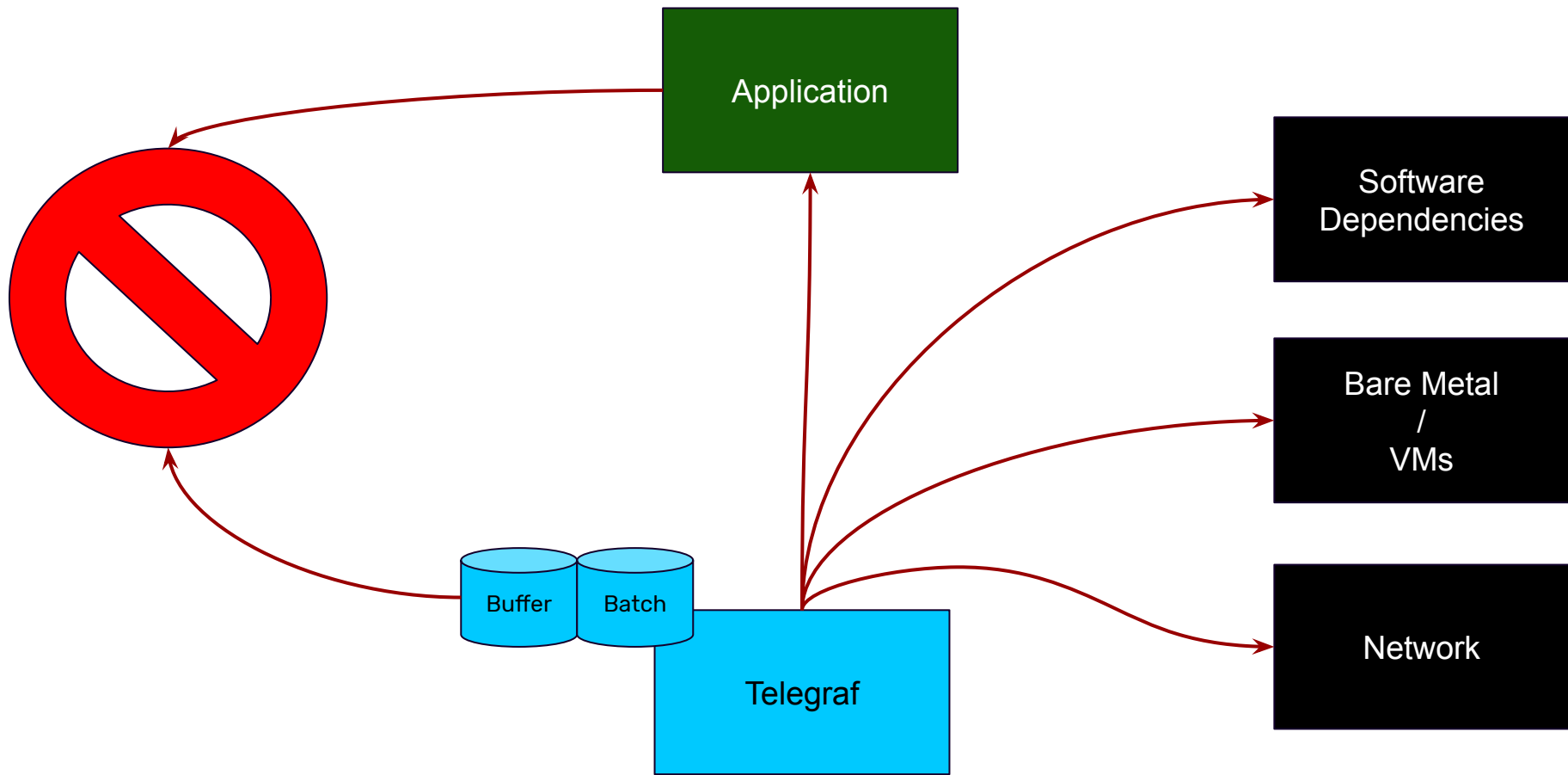
Use

`metric_buffer_limit`

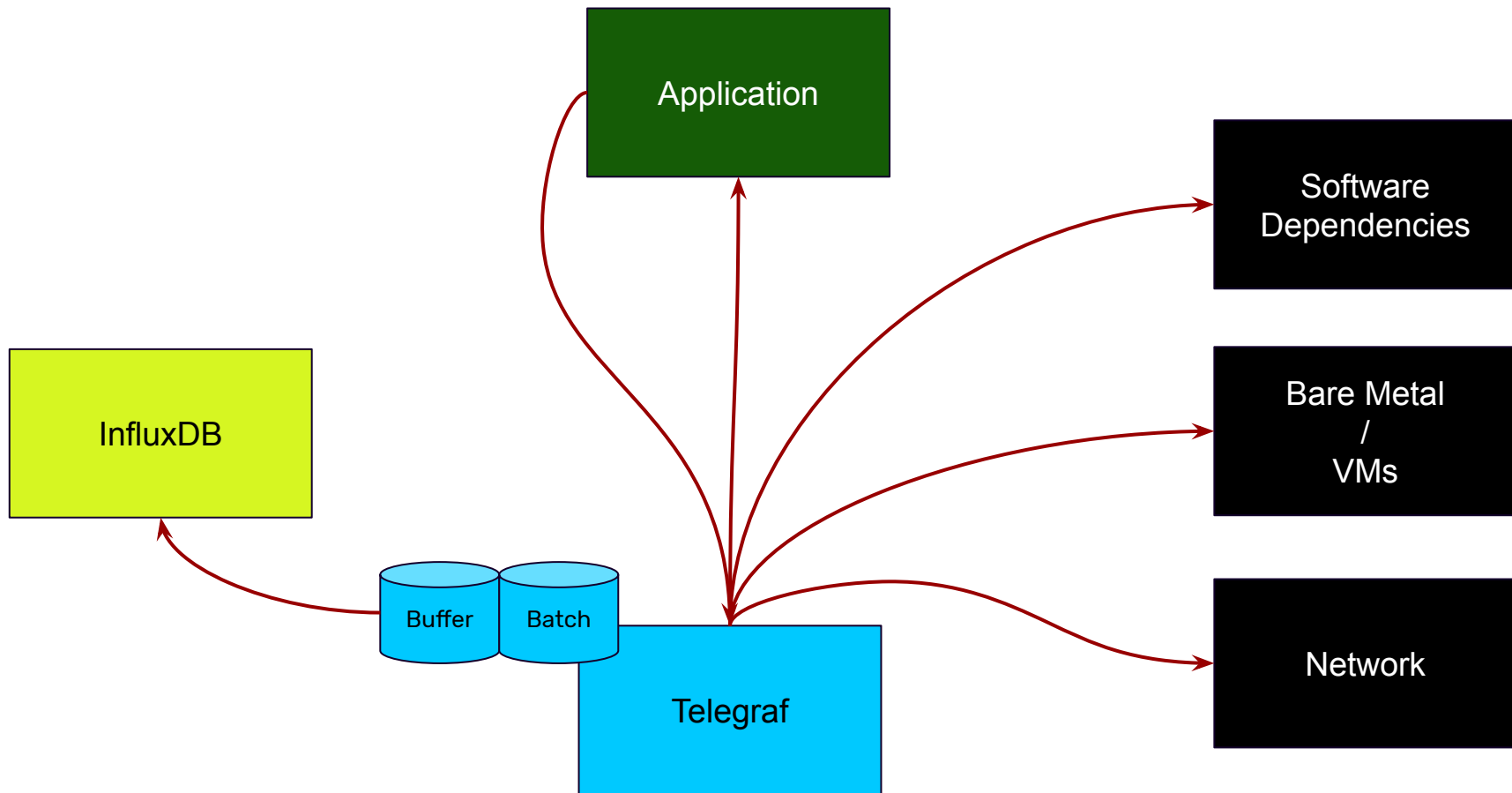
to handle downtime of your outputs











# Output Resiliency

`influxdb_listener`

`http_listener`

`v1.9 / 0.01% / It was me!`

A decorative green gradient bar with a diagonal pattern, located at the bottom of the slide.

# Output Resiliency

## influxdb\_listener

Allows Telegraf to serve as a **proxy** for the **/write** endpoint of the InfluxDB HTTP API.

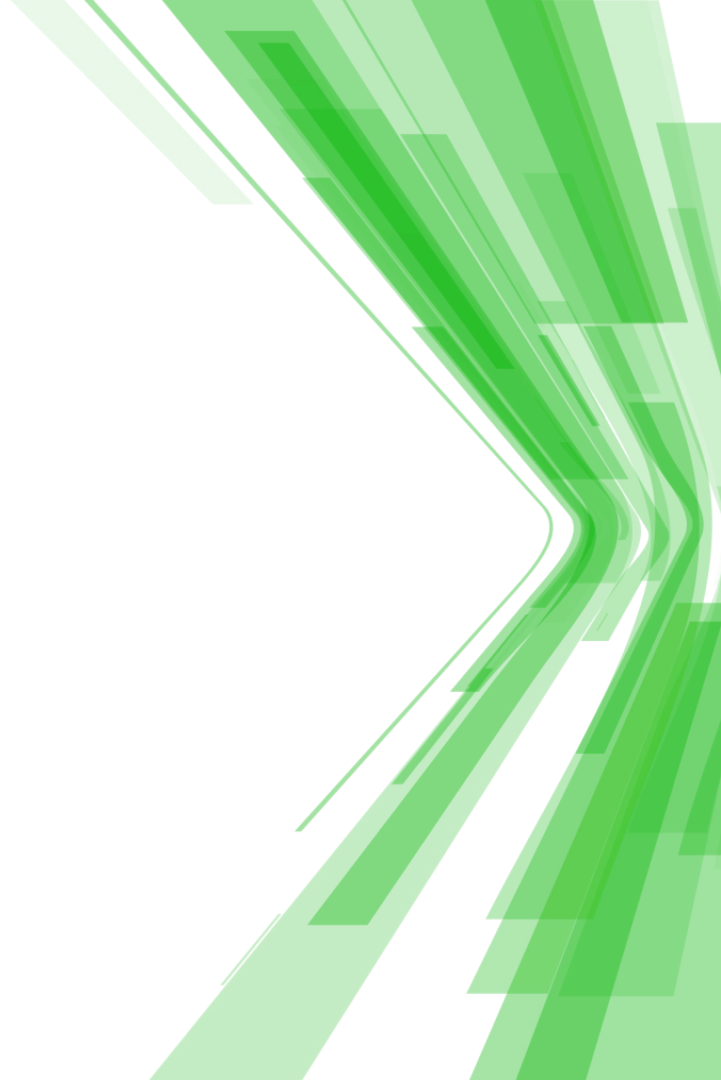
A decorative green gradient bar with a diagonal pattern, located at the bottom of the slide.

influxdb\_listener

```
[[inputs.influxdb_listener]]  
    service_address = ":8086"
```



# **v1 Client Libraries**



# Output Resiliency

`http_listener_v2`

Allows Telegraf to accept metrics over **HTTP** in **any** supported format

A decorative green gradient bar with a diagonal pattern, located at the bottom of the slide.

http\_listener\_v2

```
[[inputs.http_listener_v2]]  
  service_address = ":8080"  
  data_format = "json"
```



# JSON

[Docs](#)

```
[[inputs.http_listener_v2]]  
  data_format = "json"  
  json_name_key = "name"  
  tag_keys = ["go_version"]
```

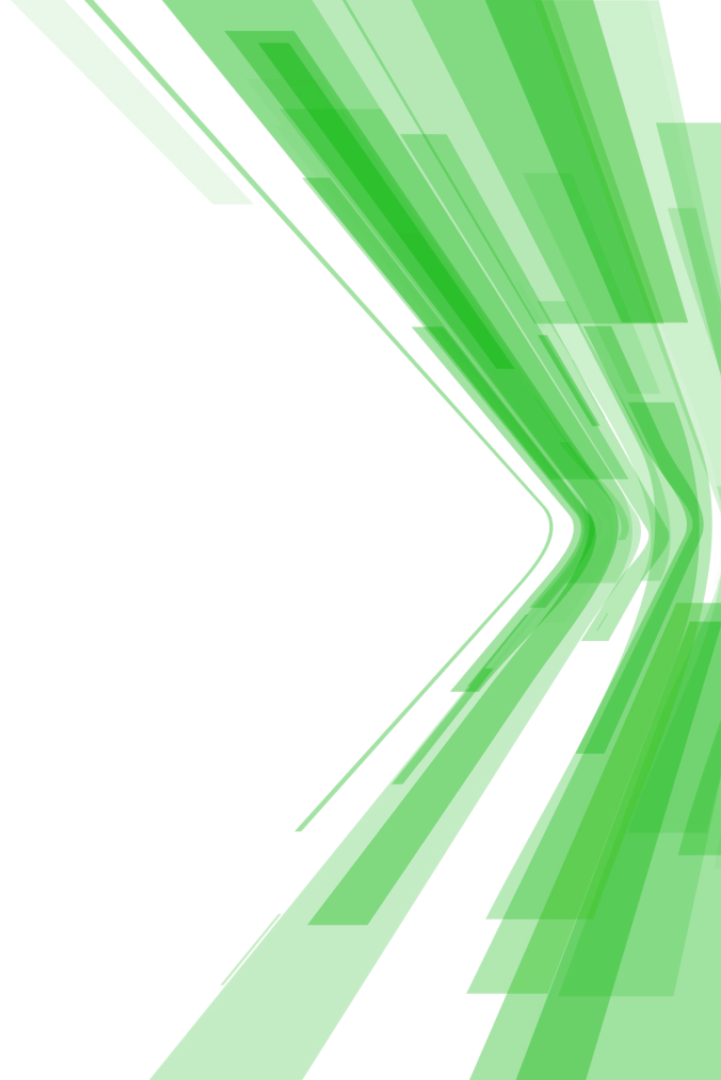
# Bring Your Own Telegraf

---

Cloud Native Telegraf

An abstract graphic composed of numerous overlapping, semi-transparent green lines and bands. These lines originate from the bottom left and fan out towards the top right, creating a sense of motion and depth. The colors range from light lime green to a vibrant, saturated green.

**Plugins come at a cost**



→ docker image ls

|             |        |
|-------------|--------|
| byot-sample | 19.3MB |
|-------------|--------|

|          |       |
|----------|-------|
| telegraf | 254MB |
|----------|-------|

# Bring Your Own Telegraf

```
FROM rawkode/telegraf:byo AS build
```

```
FROM alpine:3.7 AS telegraf
```

```
COPY --from=build /etc/telegraf /etc/telegraf
```

```
COPY --from=build /binary /bin/telegraf
```

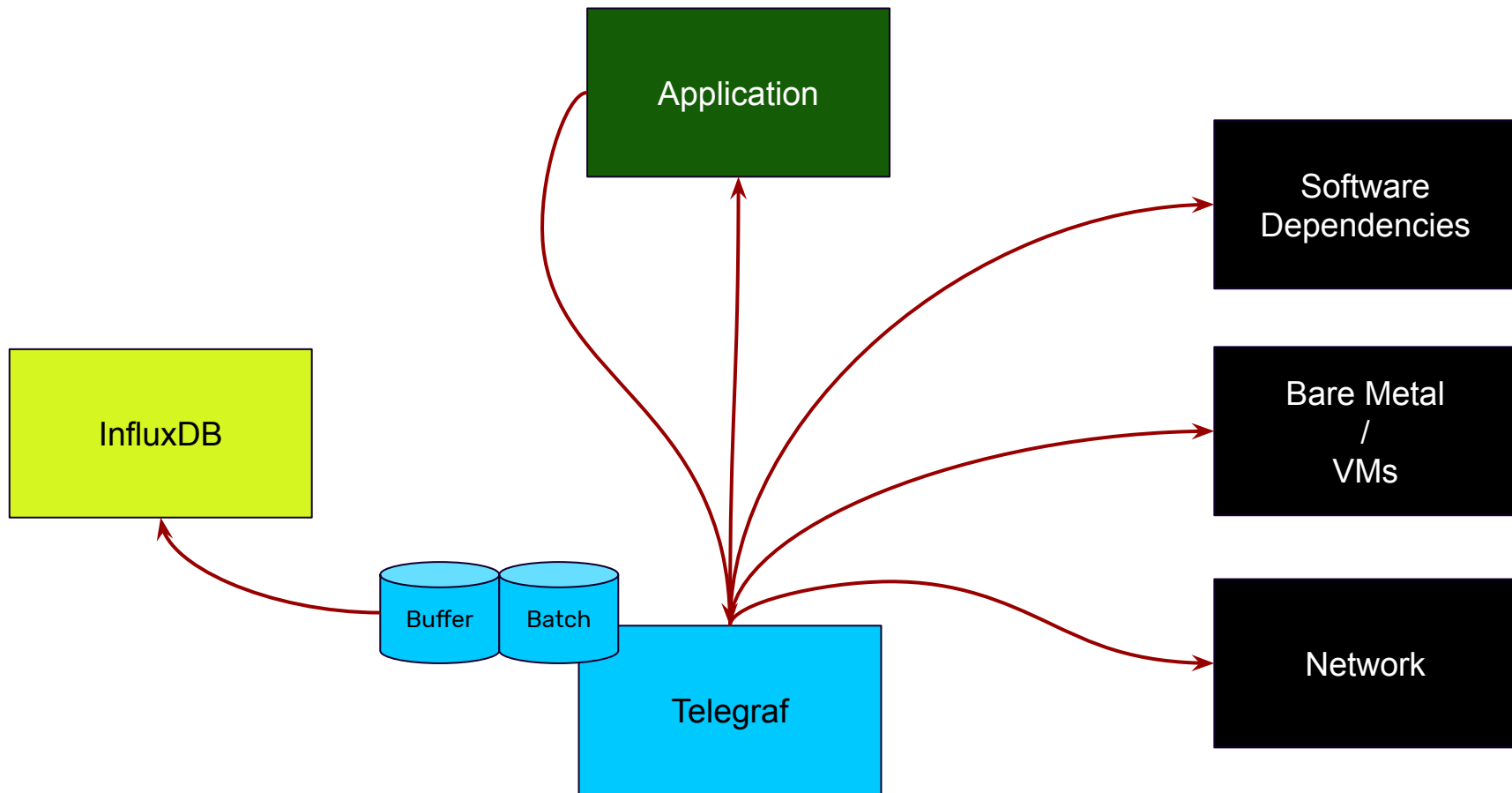
```
ENTRYPOINT [ "/bin/telegraf" ]
```

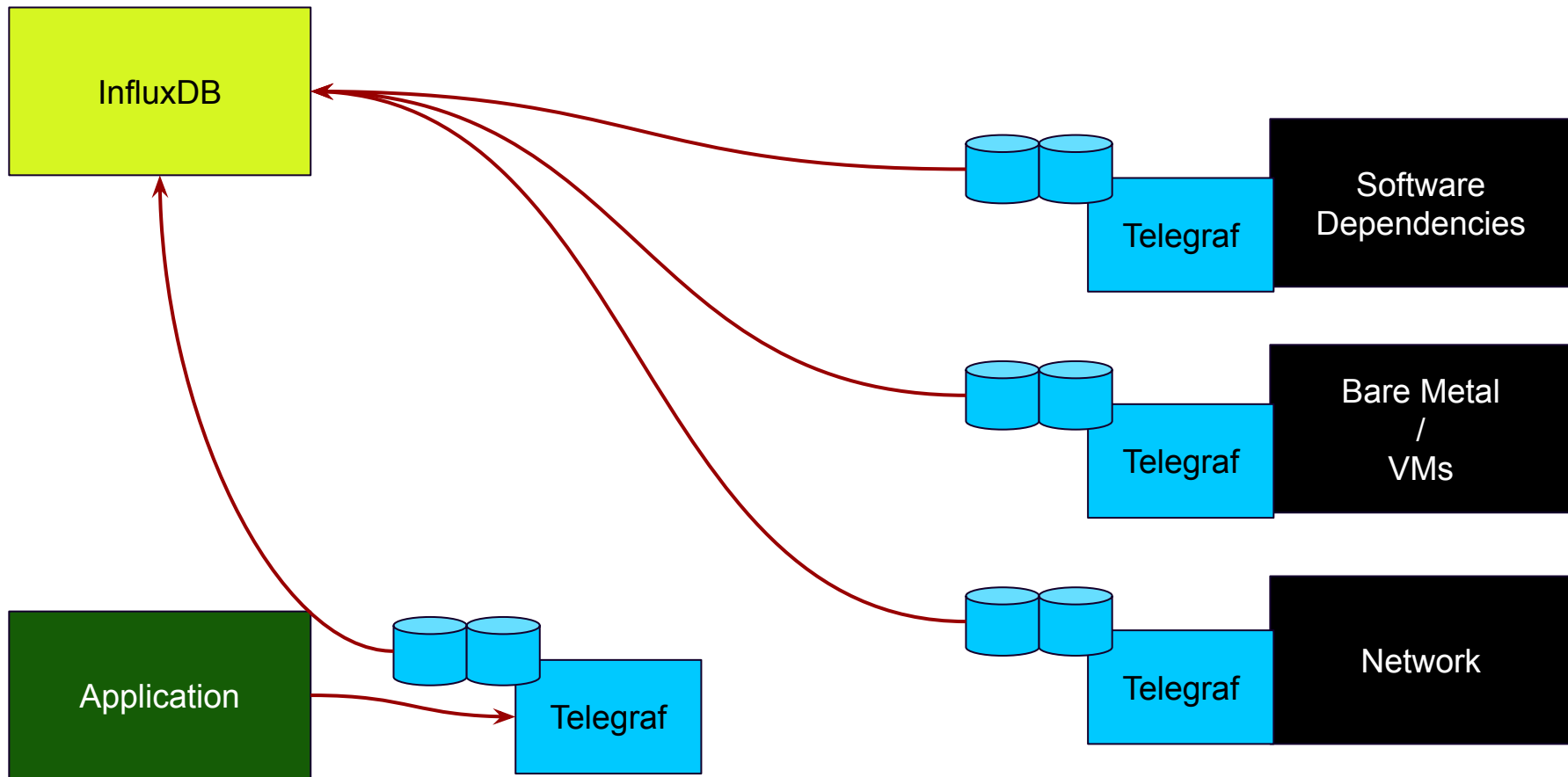
# Bring Your Own Telegraf

BYOT

BYOT Example

Click Me! Click Me!





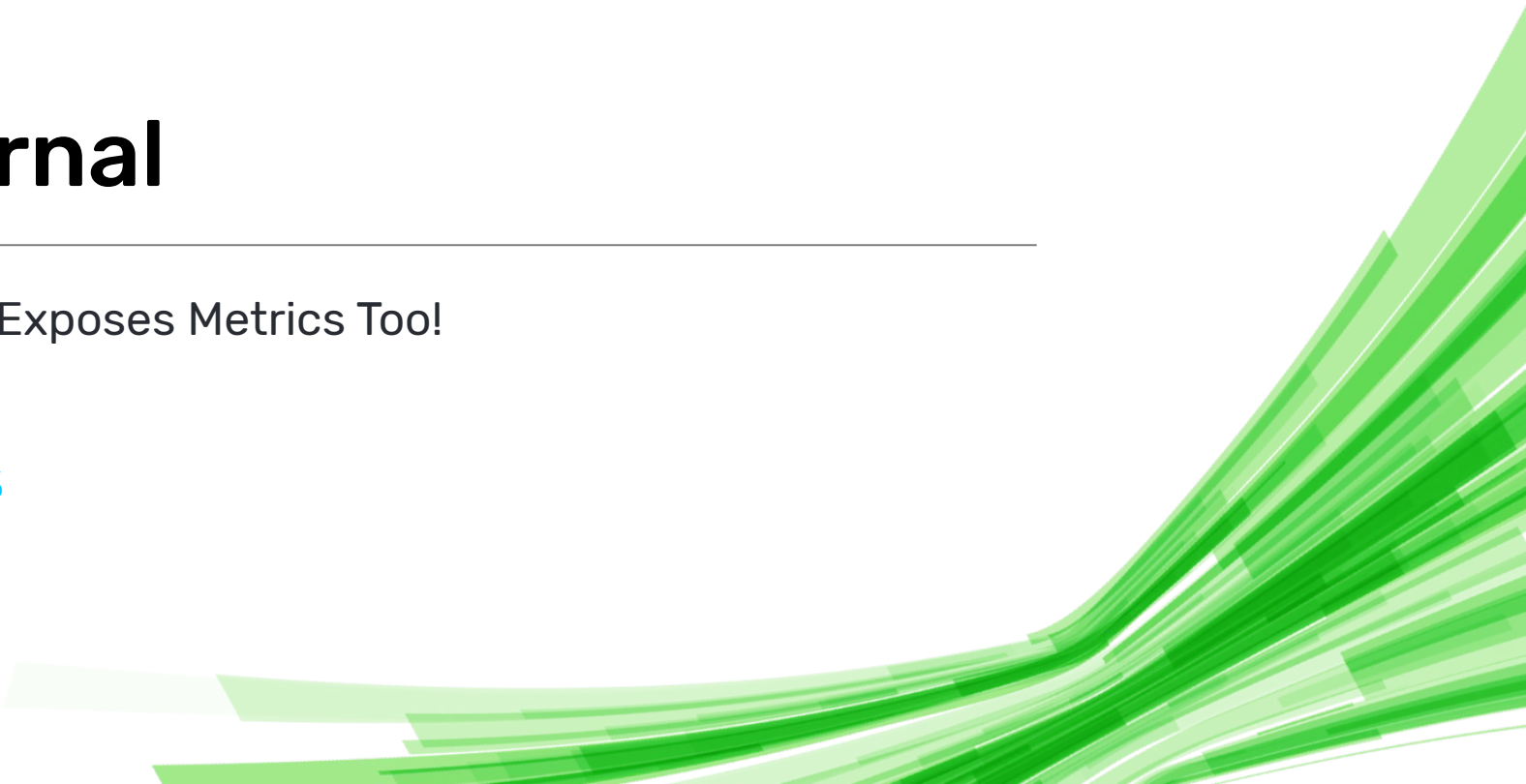


# Internal

---

Telegraf Exposes Metrics Too!

v1.2 / 0.1%

A decorative graphic on the right side of the slide, consisting of numerous overlapping, semi-transparent green lines and shapes that create a sense of motion and depth, resembling a stylized wave or a fan of lines.

# Internal

- Keep an eye on `buffer_size`
- Make sure you alert on `metrics_dropped`

# Health Output

---

Sophisticated Health Checks

v1.11 / 0%

A decorative graphic on the right side of the slide, consisting of numerous overlapping, semi-transparent green lines and shapes that create a sense of motion and depth, resembling a stylized wave or a fan of lines.

```
@app.route("/health")  
def healthcheck():  
    return "OK"
```

```
@app.route("/health")  
def healthcheck():  
    return "OK"
```



# Health Output

```
[[outputs.health]]  
  service_address = "http://:5559"  
  namepass = ["web-metrics"]
```

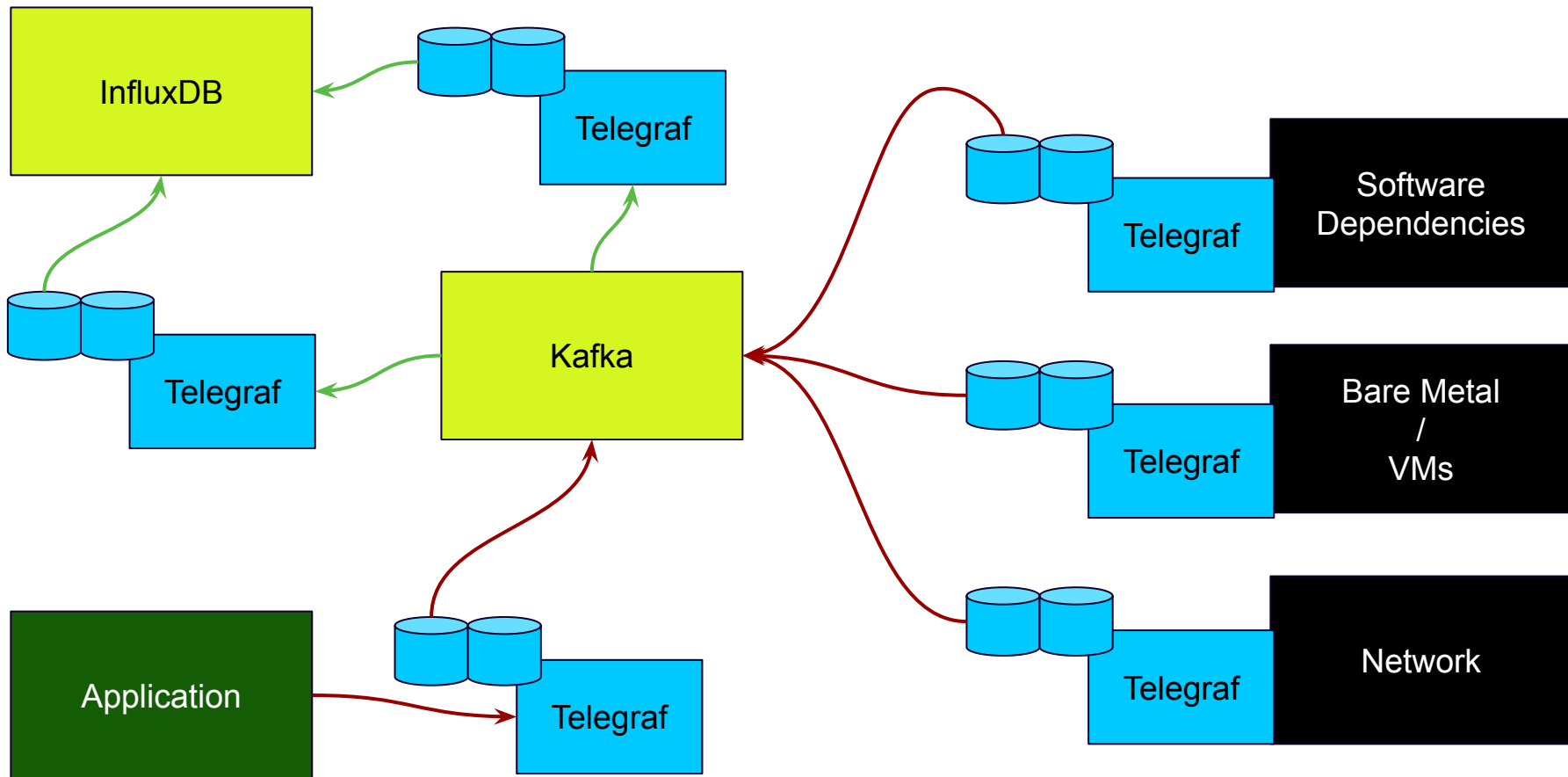
```
[[outputs.health.compares]]  
  field = "response_time"  
  lt = 0.300
```

# Kafka In, Kafka Out

---

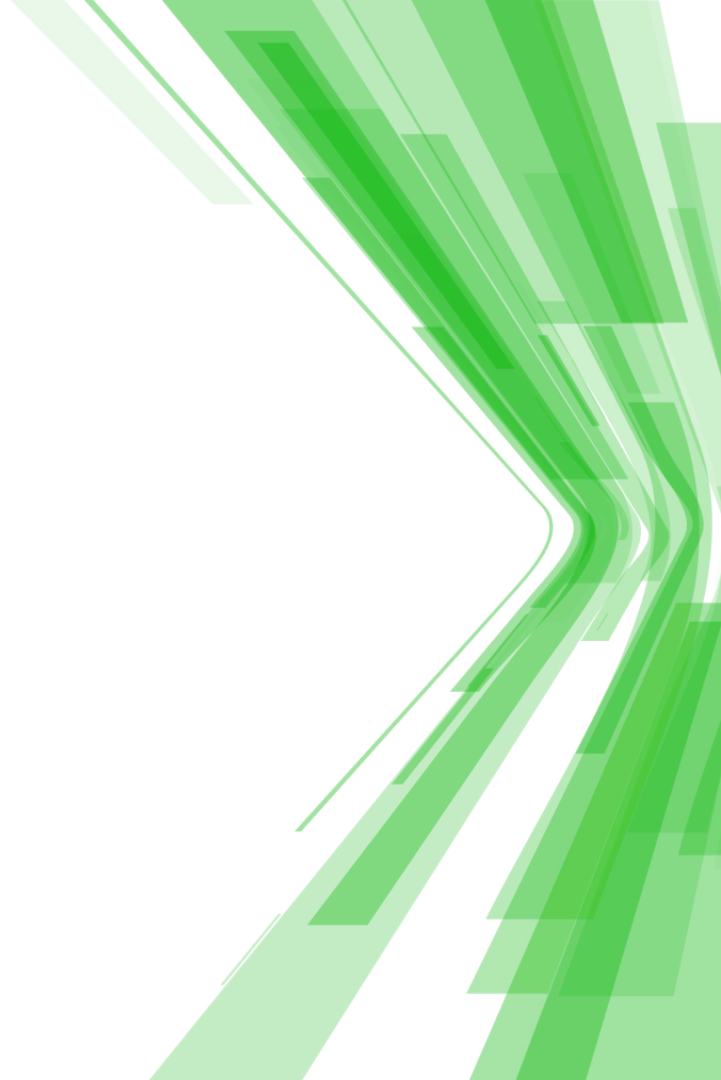
Controlling Ingress







**Demo**



**We've covered less than  
10 of the 200 plugins**





Thank you

---

@rawcode

