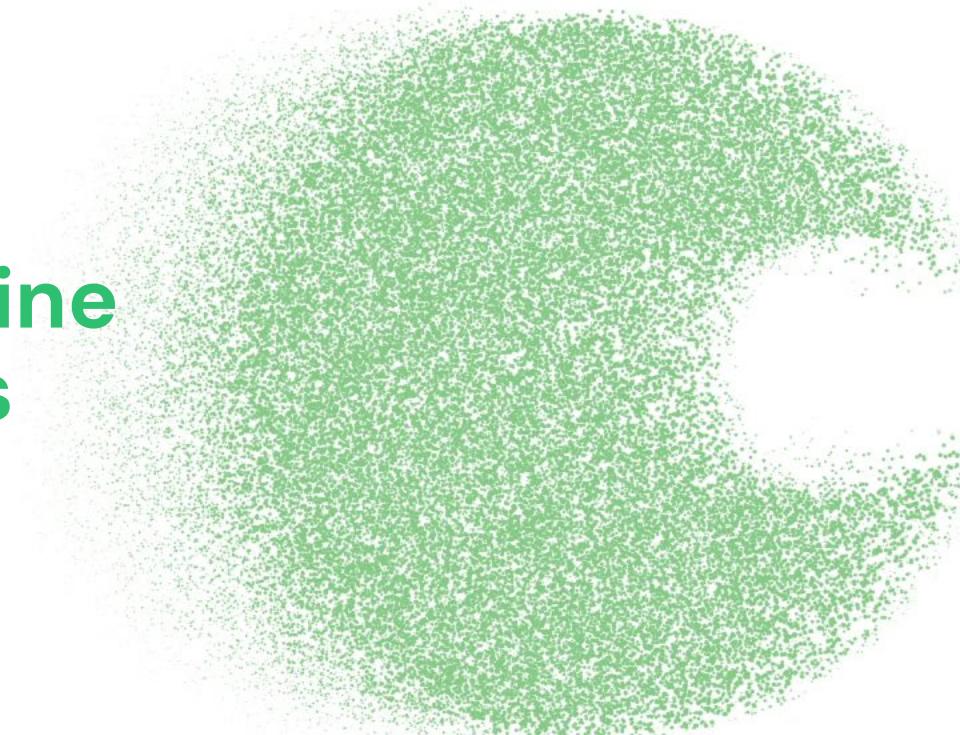


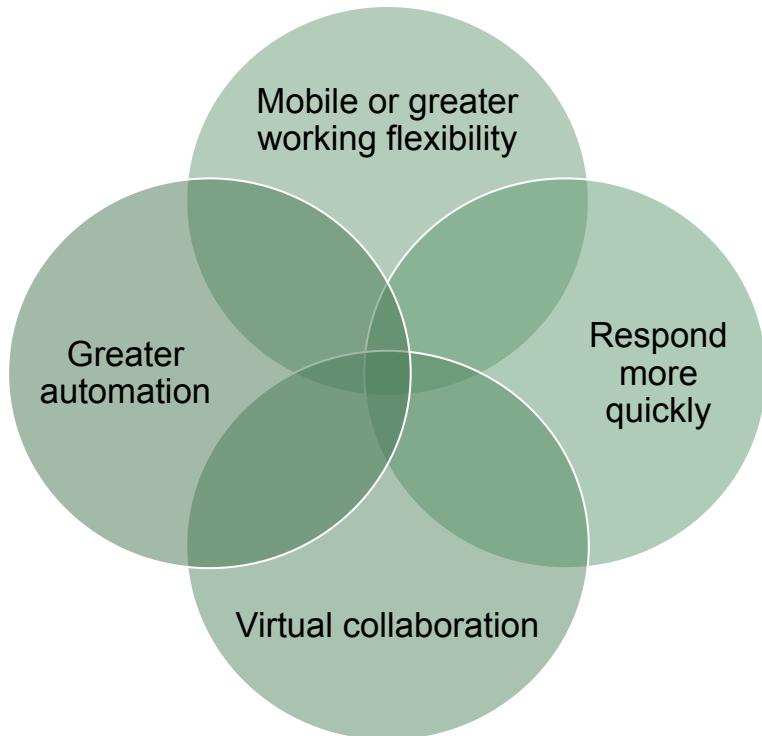
Fluent Bit, the engine to power Chat Ops

Cloud Native Rejekts
Paris 2024

Patrick Stephens & Phil Wilkins
Fluent Bit maintainer Cloud Dev Evangelist
17 Feb 2024



What & Why of ChatOps?



ChatOps – the use of chat/social channels to help Ops activities.

- Communication is key to successful Ops
- Social channels provide very effective means of communication

Some example use cases

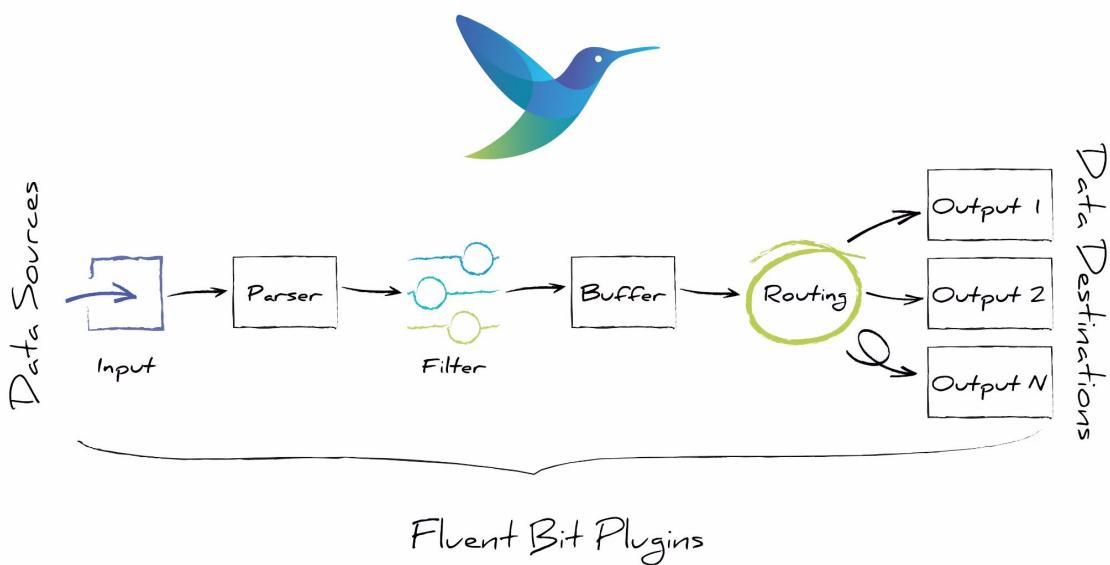
- Handle routine automatable requests
 - Credentials/Environment resets
 - Managing capacity
 - Security alerts
- Operational/Production Ops
 - Application/service faults
 - Out of hours / on-call



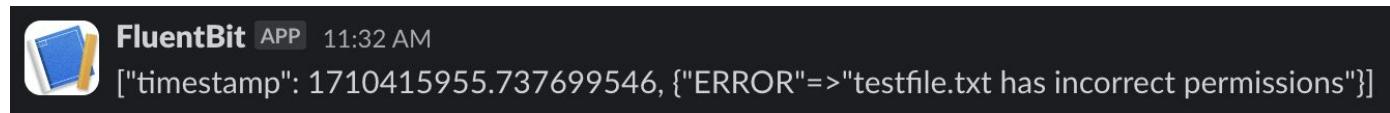
What is Fluent Bit?

Telemetry pipelines for processing logs, metrics and traces

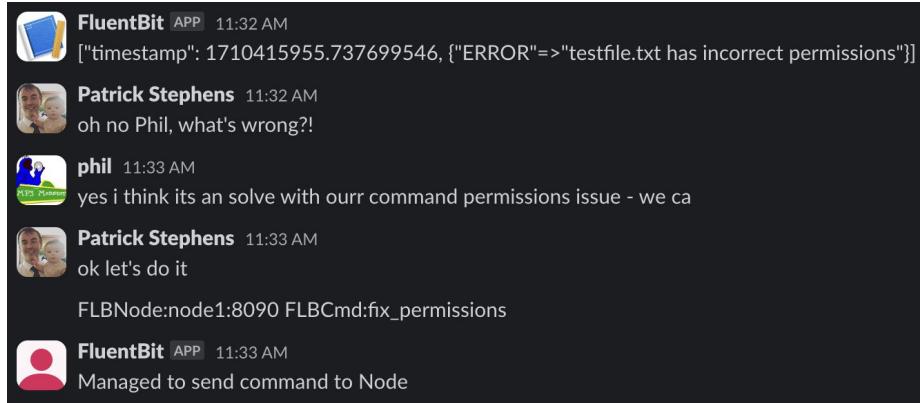
Fluent Bit Data Pipeline



How will we use Fluent Bit? Alert!



How will we use Fluent Bit? Respond!



Demo



<https://github.com/fluent/fluent-bit-chatops-demo>

LUA, the solution to all of life's problems!

Input events	Lua filter	Output events
<pre>1 {"log": "line 1"} 2 {"log": "line 2"} 3 {"log": "line 3"} 4 {"log": "line 4"} 5 {"log": "line 5"} 6 {"log": "line 6"} 7 {"log": "line 7"} 8 {"log": "line 8"} 9 {"log": "line 9"} 10 {"log": "line 10"} 11 {"log": "line 11"} 12 {"log": "line 12"} 13 {"log": "line 13"} 14 {"log": "line 14"} 15 {"log": "line 15"}</pre>	<pre>1 function cb_filter(tag, ts, record) 2 local number_start, number_end = record.log:find('%d+') 3 local number_string = record.log:sub(number_start, number_end) 4 local num = tonumber(number_string) 5 6 if num % 15 == 0 then 7 record.log = 'FizzBuzz' 8 elseif num % 5 == 0 then 9 record.log = 'Buzz' 10 elseif num % 3 == 0 then 11 record.log = 'Fizz' 12 end 13 14 return 1, ts, record 15 end</pre>	<pre>[0,{"log":"line 1"}] [0,{"log":"line 2"}] [0,{"log":"Fizz"}] [0,{"log":"line 4"}] [0,{"log":"Buzz"}] [0,{"log":"Fizz"}] [0,{"log":"line 7"}] [0,{"log":"line 8"}] [0,{"log":"Fizz"}] [0,{"log":"Buzz"}] [0,{"log":"line 11"}] [0,{"log":"Fizz"}] [0,{"log":"line 13"}] [0,{"log":"line 14"}] [0,{"log":"FizzBuzz"}]</pre>

<https://fluent.github.io/lua-sandbox>



LUA modules

```
1 # Install system dependencies
2 $ sudo apt install build-essential libssl-dev luarocks
3 # Install Lua openssl bindings
4 $ sudo luarocks install luaossl
5
6
7
```

```
1 [FILTER]
2     name      lua
3     match    *
4     script   hash.lua
5     call     cb_filter
6
7
8
```

```
1 local digest = require 'openssl.digest'
2
3 local function tohex(bytes)
4     local x = {}
5     for i=1, #bytes do
6         table.insert(x, string.format("%.2x",
7             string.byte(bytes, i)))
8     end
9     return table.concat(x, "")
10
11 local function digest_factory(algorithm)
12     return function(str)
13         local digest = digest.new(algorithm)
14         return tohex(digest:final(str))
15     end
16 end
17
18 local md5 = digest_factory('md5')
19 local sha1 = digest_factory('sha1')
20 local sha256 = digest_factory('sha256')
21
22 function cb_filter(tag, timestamp, record)
23     record.md5 = md5(record.log)
24     record.sha1 = sha1(record.log)
25     record.sha256 = sha256(record.log)
26     return 1, timestamp, record
27 end
28
29
30
```



Calyptia Core and the power of LUA!

The screenshot shows the Calyptia Core Processing Rules interface. It consists of three main sections: Input test, Actions, and Output.

Input test: A code editor containing the following JSON:

```
1 {  
2   "credit_card": "5555",  
3   "hostname": "prod",  
4   "_tag": "splunk_fw"  
5 }  
6  
7  
8  
9  
10  
11
```

Actions: A list of actions ordered by hierarchy:

- # 1 Block keys: Block credit card number (disabled)
- # 2 Custom Lua: Encode sensitive data (enabled)
- # 3 Rename keys: Rename 'hostname' key to 'h...' (enabled)
- # 4 Delete key: Rnocode sensitive data (disabled)

Output: A code editor showing the modified JSON:

```
1 {  
2   "credit_card": "****",  
3   "host": "prod"  
4 }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

Buttons: Run > (in Input test), Apply (in Actions), and Cancel (in Output).

Take control of your data and manage with ease

Reduce infrastructure complexity

Add new sources & destinations in minutes, not months

<https://core.calyptia.com/processing-rules>

The screenshot shows the Calyptia Core Processing Rules interface. At the top, there's a navigation bar with a logo, a title "Calyptia Core", and a search bar containing the URL "core.calyptia.com/processing-rules". Below the header is a main content area titled "Processing Rules". A sub-header explains that processing rules allow bringing business logic into the pipeline and setting policies. It includes sections for "Select Template" (showing "No templates available"), "Template Name" (with "Save", "Import", and "Export" buttons), and "Actions" (described as being ordered by hierarchy). The "Input" section displays a list of 10 lines of text: "line 1", "line 2", "line 3", "line 4", "line 5", "line 6", "line 7", "line 8", "line 9", and "line 10". The "Output" section shows a single line of text: "1". At the bottom, there are buttons for "Add New A..." and "Run Actions >". Status metrics at the bottom indicate "Events: 10", "Bytes: 170", "Events: 0", "Bytes: 0", and "Reduction: 0%".

Fluent Bit with Kubernetes

Phil Wilkins

ME

MANNING



billy



Questions?

Patrick Stephens

pat@chronosphere.io

<http://www.linkedin.com/in/patrickjkstephens/>

Phil Wilkins

Philip.Wilkins@Oracle.com

www.linkedin.com/in/phwilkins/

FLUENT BIT
ACADEMY



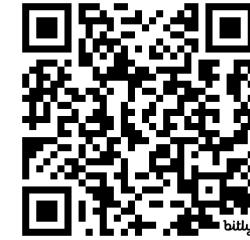
billy

Fluent Bit
Training



billy

Fluent Bit
Slack



billy

References

- Demo: <https://github.com/fluent/fluent-bit-chatops-demo>
- Recording: <https://bit.ly/fb-chatops-demo>
- Calyptia Core: <https://calyptia.com/products/calyptia-core>
- Training: <https://bit.ly/training-fluent-bit>
- Phil's book: <https://bit.ly/Fluent-Bit-Book>
- Fluent Bit: <https://fluentbit.io>
- Slack: <https://launchpass.com/fluent-all>
- LUA filter: <https://docs.fluentbit.io/manual/pipeline/filters/lua>
- LUA sandbox: <https://fluent.github.io/lua-sandbox/>
- Calyptia Core Processing Rules: <https://core.calyptia.com/processing-rules>



Training



Phil's Book

