

Grafana/Lokiの開発元に fluent-bitプラグインを 作成してフィードバックした 話

株式会社クリアコード

畑ヶ 宇宙

2020/1/15

自己紹介

株式会社クリアコード

畑ヶ 宇宙

Twitter: [@cosmo](https://twitter.com/cosmo)

GitHub: [cosmo0920](https://github.com/cosmo0920)

- 業務

- FluentdといくつかのFluentdプラグインの中の人（#fluent-bitの中の人）
- 産業用半導体メーカーのボードで動くLinuxへの現代的なブラウザの移植



fluent-bitとは

fluent-bitはFluentdファミリーのログコレクターで、C言語で書かれている。IoTを意識した軽量Fluentd。

	fluent-bit	Fluentd
実装言語	C	Ruby, C (※下回り)
依存関係	なし (依存ライブラリは同梱)	Ruby, いくつかのgem
プラグインの実装に使える言語	C, Golang (※アウトプットのみ)	Ruby

fluent-bitとは

fluent-bitはFluentdファミリーのログコレクターで、C言語で書かれている。IoTを意識した軽量Fluentd。

	fluent-bit	Fluentd
実装言語	C	Ruby, C (※下回り)
依存関係	なし (依存ライブラリは同梱)	Ruby, いくつかのgem
プラグインの実装に使える言語	C, Golang (※アウトプットのみ)	Ruby

fluent-bit Go言語 outputプラグインの作り方

```
package main
import (
    "C"
    "log"
    "unsafe"
    "github.com/fluent/fluent-bit-go/output"
)
//export FLBPluginRegister
func FLBPluginRegister(def unsafe.Pointer) int {
    return output.FLBPluginRegister(def, "myout", "Sample multiple instances plugin.")
}
//export FLBPluginInit
func FLBPluginInit(plugin unsafe.Pointer) int {
    id := output.FLBPluginConfigKey(plugin, "id")
    // Set the context to point to any Go variable
    output.FLBPluginSetContext(plugin, id)
    // do configure
    return output.FLB_OK
}
//export FLBPluginFlushCtx
func FLBPluginFlushCtx(ctx, data unsafe.Pointer, length C.int, tag *C.char) int {
    // Type assert context back into the original type for the Go variable
    id := output.FLBPluginGetContext(ctx).(string)
    log.Printf("[multiinstance] Flush called for id: %s", id)
    // do something with multi instance contexts

    return output.FLB_OK
}
//export FLBPluginExit
func FLBPluginExit() int {
    return output.FLB_OK
}
func main() {
```

FLBPluginFlushCtxを
使うと複数インスタンス
を管理するfluent-bitプラ
グインにできる

詳しい解説は <https://www.clear-code.com/blog/2019/7/24.html> を参照。

fluent-bit Go言語 outputプラグインの作り方

1. 共有ライブラリとしてビルド

```
$ go build --buildmode=c-shared myout.so my_out_plugin.go
```

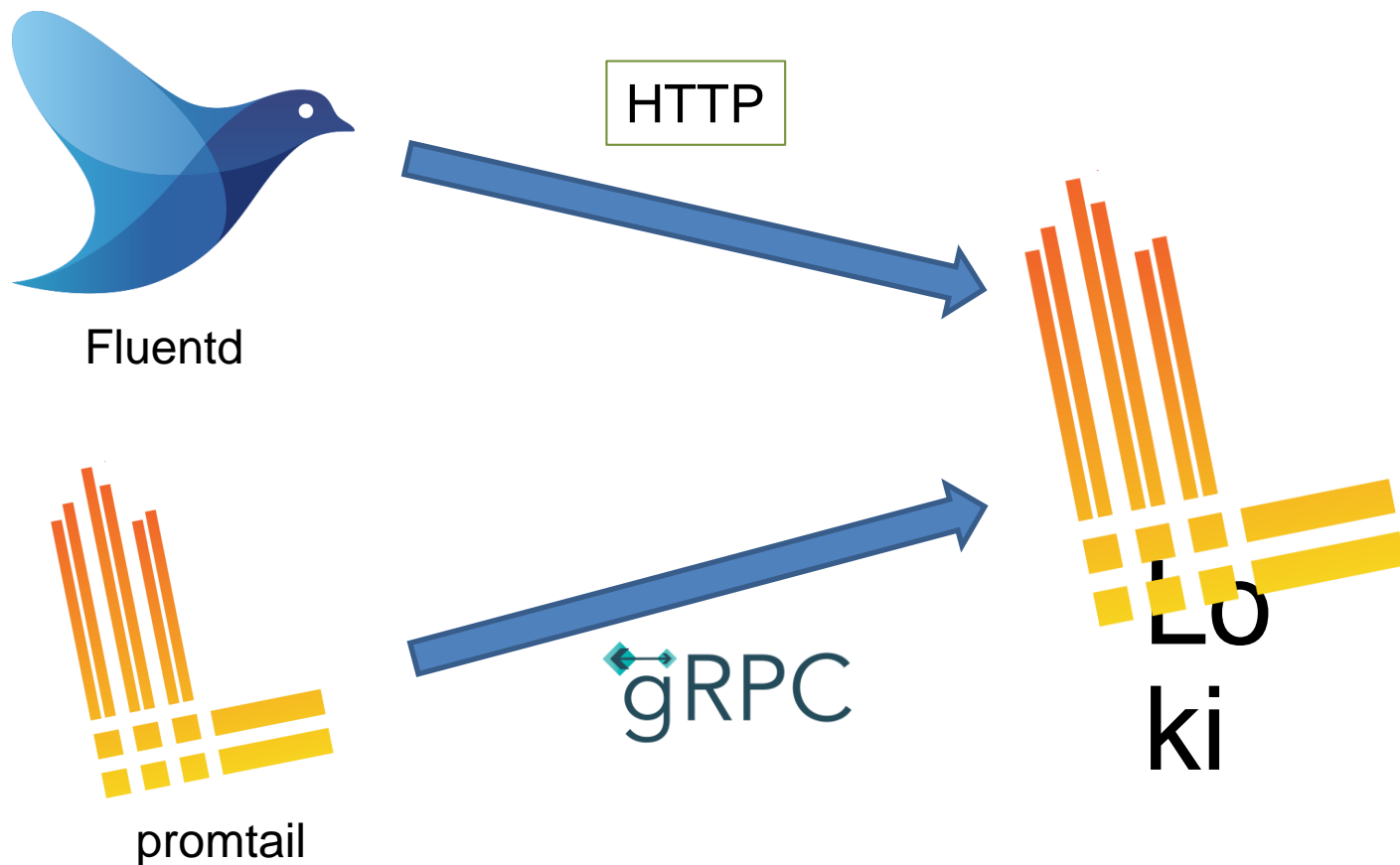
2. 読み込みパスに追加して起動

```
$ fluent-bit -e /path/to/myout.so -i cpu -o myout
```

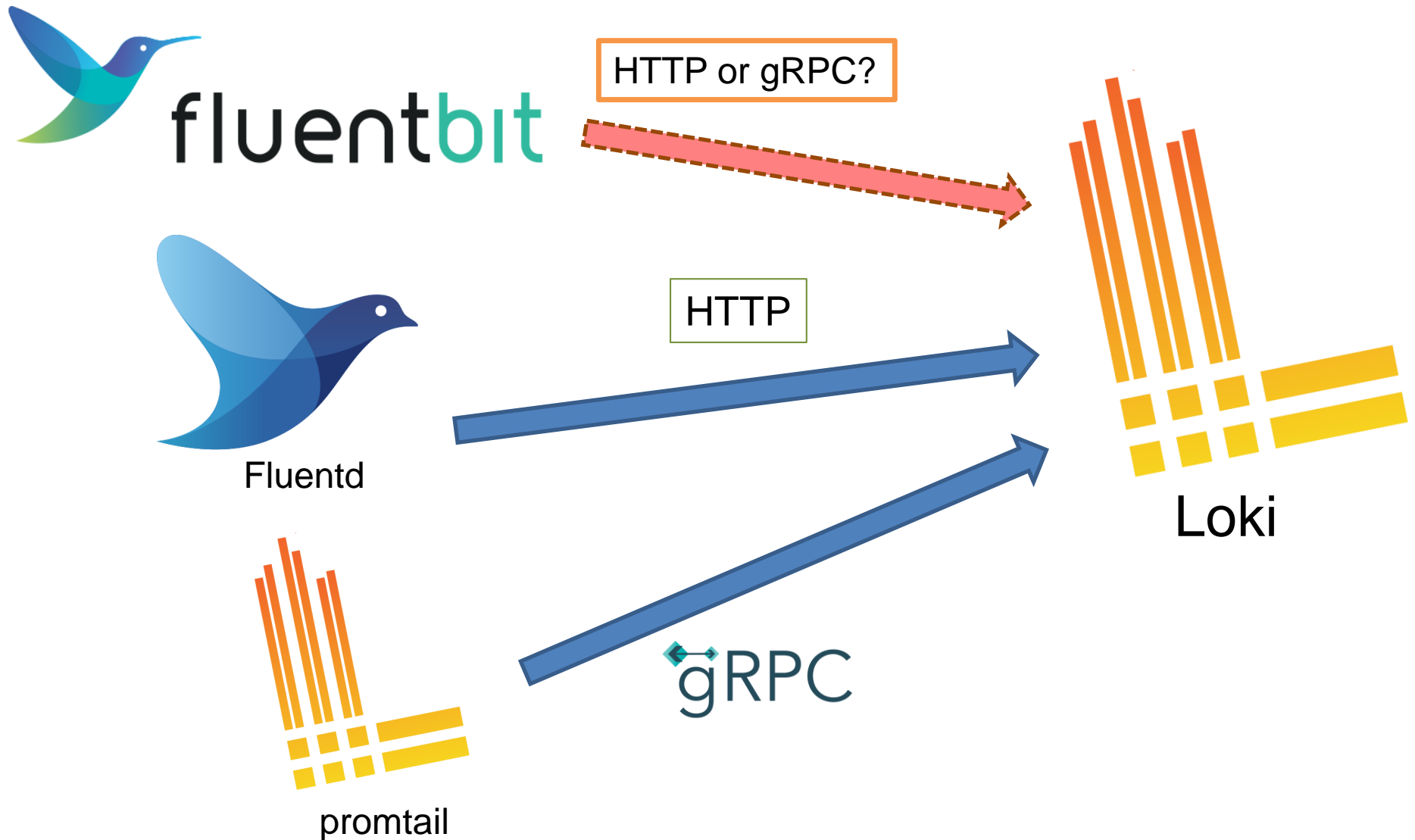
または

```
$ cp myout.so /usr/lib/x86_64-linux-gnu/  
$ fluent-bit -i cpu -o myout
```

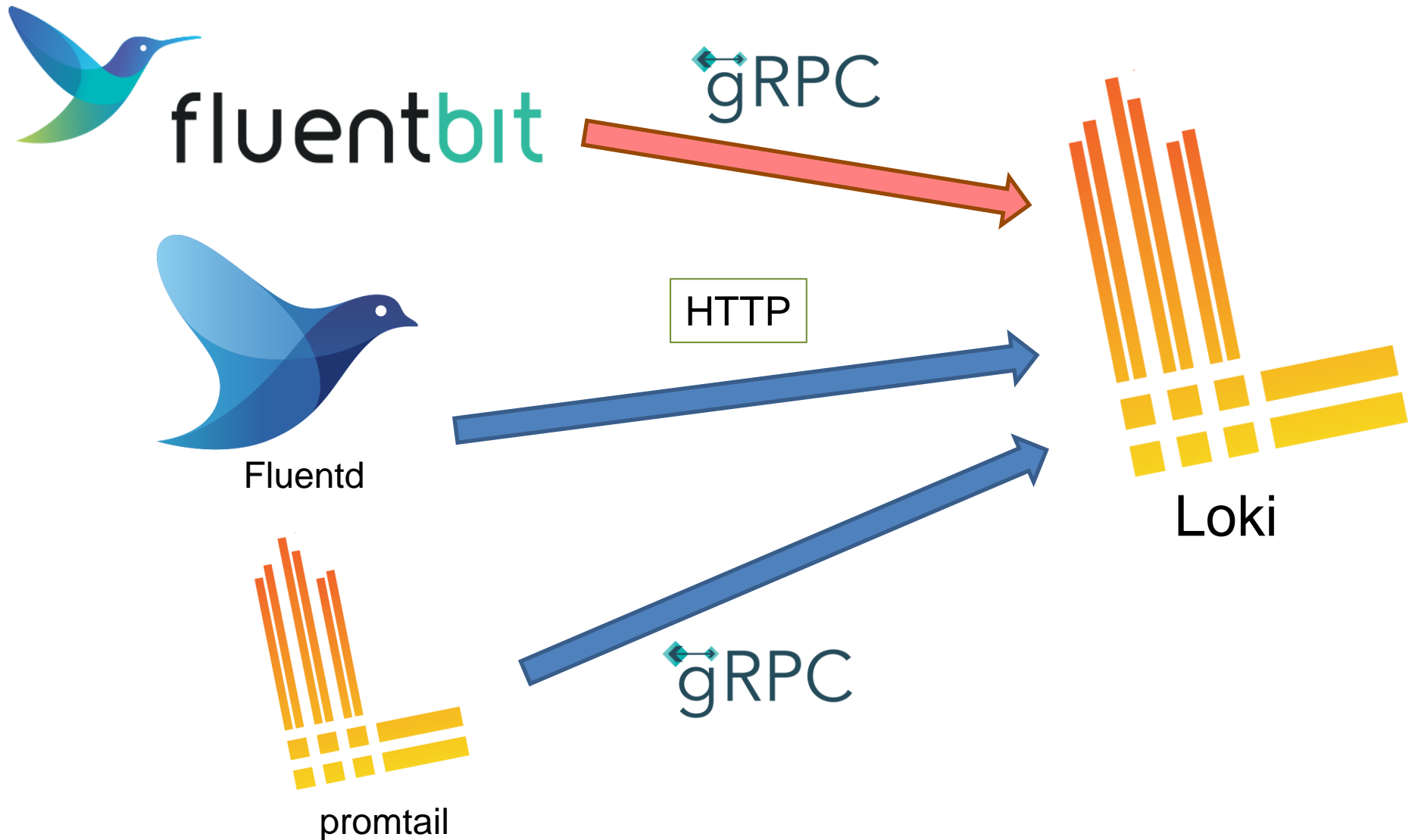
Grafana/Lokiにログを転送するには



Grafana/Lokiにログを転送するには(2)



Grafana/Lokiにログを転送するには(3)



GolangでLokiにログを送るには



```
package main
```

```
import "github.com/grafana/loki/pkg/promtail/client"
import "github.com/sirupsen/logrus"
import kit "github.com/go-kit/kit/log/logrus"
import "github.com/cortexproject/cortex/pkg/util/flagext"
import "github.com/prometheus/common/model"
import "time"
```

```
func main() {
    cfg := client.Config{}
    // Init everything with default values.
    flagext.RegisterFlags(&cfg)
    var clientURL flagext.URLValue

    url := "http://localhost:3100/api/prom/push"
    // Override some of those defaults
    _ = clientURL.Set(url)
    cfg.URL = clientURL
    cfg.BatchWait = 1
    cfg.BatchSize = 10 * 1024
    log := logrus.New()
    loki, _ := client.New(cfg, kit.NewLogrusLogger(log))
    line := `{"message": "Sent from Golang!"}`
    labelValue := "from-golang"
    labelSet := model.LabelSet{"lang": model.LabelValue(labelValue)}
    _ = loki.Handle(labelSet, time.Now(), line)
    // Ensure to send record into Loki.
    time.Sleep(3 * time.Second)
}
```

Promtailの
クライアントを作
成

Promtailの
クライアントを
使ってログを送信

詳しい解説は <https://www.clear-code.com/blog/2019/7/31.html> を参照。

GolangでLokiにログを送るには（要約）

gRPCで送るには

Promtailのクライアントのインスタンスを作成

PromtailのクライアントAPIを使ってログを送信

gRPCのAPIは**非同期**

`loki.Handle(labelSet, time.Now(), line)`を実行してもすぐにはLokiには送信されない。バッチ的に送られる

fluent-bitのGo製Lokiプラグイン

作ってみた。

- [cosmo0920/fluent-bit-go-loki](https://github.com/cosmo0920/fluent-bit-go-loki)
 - fluent-bitのGolangインターフェースを使ったGolang製のfluent-bitプラグイン
 - Linux/macOS/Windowsで動作する

fluent-bitのGo製Lokiプラグイン

- 作った理由は？
 - 同じログコレクターの中の人としてLokiに触れておきたい
 - Promtailよりも慣れ親しんだプロトコルでLokiにログをいれてみたい
 - ~~Promtailがちょっと使いづらい~~

ある日。。。。



grafana / loki

Watch 196 Star 8.4k Fork 674

Code Issues 110 Pull requests 21 Actions Projects 0 Wiki Security Insights

Support fluent-bit also in addition/replacement for fluentd

New issue

#770

Closed rverma-nikiyai opened this issue on 17 Jul 2019 · 8 comments



rverma-nikiyai commented on 17 Jul 2019 · edited

Is your feature request related to a problem? Please describe.
Fluentd is still a resource hoggy service, also multiple elk/splunk etc consumers prefers fluent-bit.

Describe the solution you'd like
We should support fluent-bit also, there is some initial work as well <https://github.com/cosmo0920/fluent-bit-go-loki>
Additionally, fluent-bit native support for kubernetes allows gathering all labels/annotations by default.
Can we manage this officially at <https://github.com/fluent/fluent-bit/tree/master/plugins>



3



cyriltovena added **component/agent** **component/integrations** **help wanted** **kind/feature** labels on 17 Jul 2019



cyriltovena commented on 17 Jul 2019

Contributor +

A big yes !



rverma-nikiyai mentioned this issue on 18 Jul 2019

Store applications persist logs (human-readable) to disk or S3 Storage. #766

Closed



cyriltovena commented on 2 Aug 2019

Contributor +

@cosmo0920 would you be interested into merging this into our repository ?

Assignees

No one assigned

Labels

component/agent

component/integrations

help wanted

keepalive

kind/feature

Projects

None yet

Milestone

No milestone

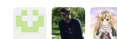
Notifications

Customize

Unsubscribe

You're receiving notifications because you were mentioned.

3 participants



Grafana/Lokiの中の人から打診



cyriltovena commented on 2 Aug 2019

Contributor



@cosmo0920 would you be interested into merging this into our repository ?

- Grafana/Lokiの中の人から作成していたfluent-bit-go-lokiをGrafana/Lokiにマージしてみるのはどうよ？という打診が来た。

フィードバックする前にいくつかやり取り



cosmo0920 commented on 2 Aug 2019 • edited ▾

Contributor + 😊 ...

Yeah. I'm interested in merging my fluent-bit go loki plugin here.

But, I have questions to start to work:

- How do I manage fluent-bit-go-loki plugin's library dependencies?
- I want to continue to improve fluent-bit-go-loki plugin (most of my hobby work, not Fluentd/fluent-bit contributor work). How should I support to improvement here?
- Need to rectify [loki's main Makefile](#) for this work?



cyriltovena commented on 2 Aug 2019 • edited ▾

Contributor + 😊 ...

I think the ultimate goal will be to have an official plugin in the fluentbit repository. But let's fist incubate a version here. I'll get in touch with the fluentbit team see how that would work. I'm not sure if the a go version can become official. Looks like most officials one are written in C.

How do I manage fluent-bit-go-loki plugin's library dependencies?

Right now we use dep. You can create a new `cmd/fluent-bit` folder.

How should I support to improvement here?

You would still be able to contribute here, though you would need review first by us. However I'm working on a governance document and you could become a member of this Repository. You would still need review that how we manage this repo. Basically decisions will be taken together as a team.

Need to rectify loki's main Makefile for this work?

Yes and we're here to help you for this. I think you should not worry too much about this for now, let's start step by step. I think it would be easier to contribute one small PR at the time:

- Fluentbit plugin code
- documentation
- Makefile & release.

- フィードバックに興味があることと、継続して関わっていききたい旨を意思表示

フィードバックしてみた



fluent-bit shared object go plugin #847

Edit

Merged cyriltovena merged 48 commits into grafana:master from cosmo0920:fluent-bit-shared-object-go-plugin on 27 Sep 2019

Conversation 73 Commits 48 Checks 1 Files changed 97 +32,407 -13



cosmo0920 commented on 5 Aug 2019 • edited

Contributor + 🗨️ ⋮

What this PR does / why we need it:

The first step to merge <https://github.com/cosmo0920/fluent-bit-go-loki>.

Which issue(s) this PR fixes:

No fixes. This is the first task of #770.

Special notes for your reviewer:

I think that we would have to create `fluent-bit/fluent-bit-go-loki` folder instead of `cmd/fluent-bit`. Because fluent-bit-go-loki Golang plugin uses shared object format. It is not executable.

Building steps for fluent-bit-go-loki

fluent-bit-go-loki can build with:

```
$ (cd fluent-bit/fluent-bit-go-loki && go build -buildmode=c-shared -o out_loki.so .)
```

Or, make with added Makefile(building and testing):

```
$ (cd fluent-bit/fluent-bit-go-loki && make)
```

Testing for fluent-bit-go-loki

fluent-bit-go-loki test can execute with:

```
$ (cd fluent-bit/fluent-bit-go-loki && go test -cover -race -coverprofile=coverage.txt -covermode=)
```

Or, make with added Makefile:

```
$ (cd fluent-bit/fluent-bit-go-loki && make test)
```

Checklist

- ☒ Documentation added
- ☒ Tests updated

👍 2 🐞 1

Reviewers

cyriltovena

sh0rez

joe-elliott

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Customize

🔊 Unsubscribe

You're receiving notifications because you were mentioned.

6 participants



☐ Allow edits from maintainers.

[Learn more](#)

- [Grafana/Loki#847](#)
- 一般にOSSにフィードバックする作法に準じればちゃんと見てもらえる
 - Pull requestテンプレートを埋めて書き込めばOK
 - 独自フォーマットで説明するのはよほどのことがない限り避けた方がよい

フィードバックしてみた (2)



fluent-bit: Support multi instances for fluent-bit go loki plugin (revised) #1454

Edit

Closed cosmo0920 wants to merge 1 commit into grafana:master from cosmo0920:support-multi-instances

Conversation 4 Commits 1 Checks 1 Files changed 1 +46 -14



cosmo0920 commented 21 days ago • edited

Contributor + 😊 ...

I've noticed that current implementation does not cover multi instance use case.

Signed-off-by: Hiroshi Hatake cosmo0920.oucc@gmail.com

What this PR does / why we need it:

In the previous implementation, this fluent-bit go loki plugin does not support multi instances.

Because global variable which is `var plugin *loki` is used for plugin instance management.

So, it is always overridden when loki plugin is used.

This implementation is based for [fluent-plugin-go-s3](#) plugin multi instances support code.

Also, I know the similar patch is sent in [#1294](#), but this patch needn't specify context id by users. This ID should be *calculated automatically*. Users can use fluent-bit plugin in the same way after applying this patch.

Which issue(s) this PR fixes:

Fixes [#1446](#)

Special notes for your reviewer:

Built binary with `go build -race` and this patch causes SEGV.

But this SEGV occurs current master with `go build -race`: [dafb9d8](#)

My used fluent-bit config is:

```
[INPUT]
  Name cpu
```

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Customize

🔊 Unsubscribe

You're receiving notifications because you authored the thread.

3 participants



☐ Allow edits from maintainers.

[Learn more](#)

- [Grafana/Loki#1454](#)
- 複数インスタンスサポートの提案

フィードバックしてみた (2)

ClearCode



JensErat commented 18 days ago

Author Contributor + 🗨️ ...

This is an updated version, merging ideas of @cosmo0920 in #1454 into my work. The fluent-bit side configuration of an ID is indeed not required, but the code can be simplified even further by passing the plugin reference as pointer, not the ID. I was able to remove a whole bunch of lines.

I had a shot at adding some kind of end-to-end test, but this is not a trivial task. cgo and golang tests do not really go together well, I have to provide some mocks and function wrappers (in a totally reasonable amount, but still work to do). At least for the receiver side there is reusable code in promtail. I will not be able to finish this within the next days though, out of office again for more than a week now. I totally agree we should have tests, but I'd love to see this split apart to another follow-up PR since this blocks at least us and @fmax in #1446 from using upstream releases.

I will have another go on the test code as soon as time allows, but for now I just have some vast ideas on how to test the plugin.

Unlike cosmo's PR, this also keeps the ID reference for logs. This makes understanding what's going wrong much easier. For example (see the `id` log label):

```
level=info caller=out_loki.go:56 id=0 [flb-go]="Starting fluent-bit-go-loki" version="(version=f:
level=info caller=out_loki.go:58 id=0 [flb-go]="provided parameter" URL=http://loki-distributor.
level=info caller=out_loki.go:59 id=0 [flb-go]="provided parameter" TenantID=platform
level=info caller=out_loki.go:60 id=0 [flb-go]="provided parameter" BatchWait=5s
level=info caller=out_loki.go:61 id=0 [flb-go]="provided parameter" BatchSize=32000
level=info caller=out_loki.go:62 id=0 [flb-go]="provided parameter" Labels="[job=fluent-bit]"
level=info caller=out_loki.go:63 id=0 [flb-go]="provided parameter" LogLevel=info
level=info caller=out_loki.go:64 id=0 [flb-go]="provided parameter" AutoKubernetesLabels=false
level=info caller=out_loki.go:65 id=0 [flb-go]="provided parameter" RemoveKeys=[]
level=info caller=out_loki.go:66 id=0 [flb-go]="provided parameter" LabelKeys=[HOSTNAME]
level=info caller=out_loki.go:67 id=0 [flb-go]="provided parameter" LineFormat=0
level=info caller=out_loki.go:68 id=0 [flb-go]="provided parameter" DropSingleKey=true
level=info caller=out_loki.go:69 id=0 [flb-go]="provided parameter" LabelMapPath=map[]
level=info caller=out_loki.go:56 id=1 [flb-go]="Starting fluent-bit-go-loki" version="(version=f:
level=info caller=out_loki.go:58 id=1 [flb-go]="provided parameter" URL=http://loki-distributor.
level=info caller=out_loki.go:59 id=1 [flb-go]="provided parameter" TenantID=audit
level=info caller=out_loki.go:60 id=1 [flb-go]="provided parameter" BatchWait=5s
level=info caller=out_loki.go:61 id=1 [flb-go]="provided parameter" BatchSize=32000
level=info caller=out_loki.go:62 id=1 [flb-go]="provided parameter" Labels="[job=fluent-bit]"
level=info caller=out_loki.go:63 id=1 [flb-go]="provided parameter" LogLevel=info
level=info caller=out_loki.go:64 id=1 [flb-go]="provided parameter" AutoKubernetesLabels=false
level=info caller=out_loki.go:65 id=1 [flb-go]="provided parameter" RemoveKeys=[]
level=info caller=out_loki.go:66 id=1 [flb-go]="provided parameter" LabelKeys=[TENANT]
level=info caller=out_loki.go:67 id=1 [flb-go]="provided parameter" LineFormat=0
level=info caller=out_loki.go:68 id=1 [flb-go]="provided parameter" DropSingleKey=true
level=info caller=out_loki.go:69 id=1 [flb-go]="provided parameter" LabelMapPath=map[]
[snip]
level=error caller=client.go:236 id=1 component=client host=loki-distributor.logging.svc:3100 ms[
```

- [Grafana/Loki#1454](#)
- 複数インスタンスサポートの提案
- 結果的には直接はマージされなかった。アイディアは取り込まれた。



cyriltovena approved these changes 8 days ago

[View changes](#)

FluentBitのLokiプラグインの設定

- fluent-bitの設定ファイルはINIファイル形式。
 - 下記の設定はCPU使用率を取得し、Lokiに送る

[INPUT]

```
Name cpu  
Tag my_cpu
```

[OUTPUT]

```
Name loki  
Match my_cpu  
Url http://localhost:3100/loki/api/v1/push  
BatchWait 1  
BatchSize 30720  
Labels {test="fluent-bit-go", tag="my_cpu"}  
LogLevel info
```

その他の設定はGrafana/Lokiの[fluent-bitプラグインのドキュメント参照](#)してください。

Grafana/Lokiのfluent-bitプラグイン

1. 共有ライブラリとしてビルド

```
$ make fluent-bit-plugin
```

2. 読み込みパスに追加して起動

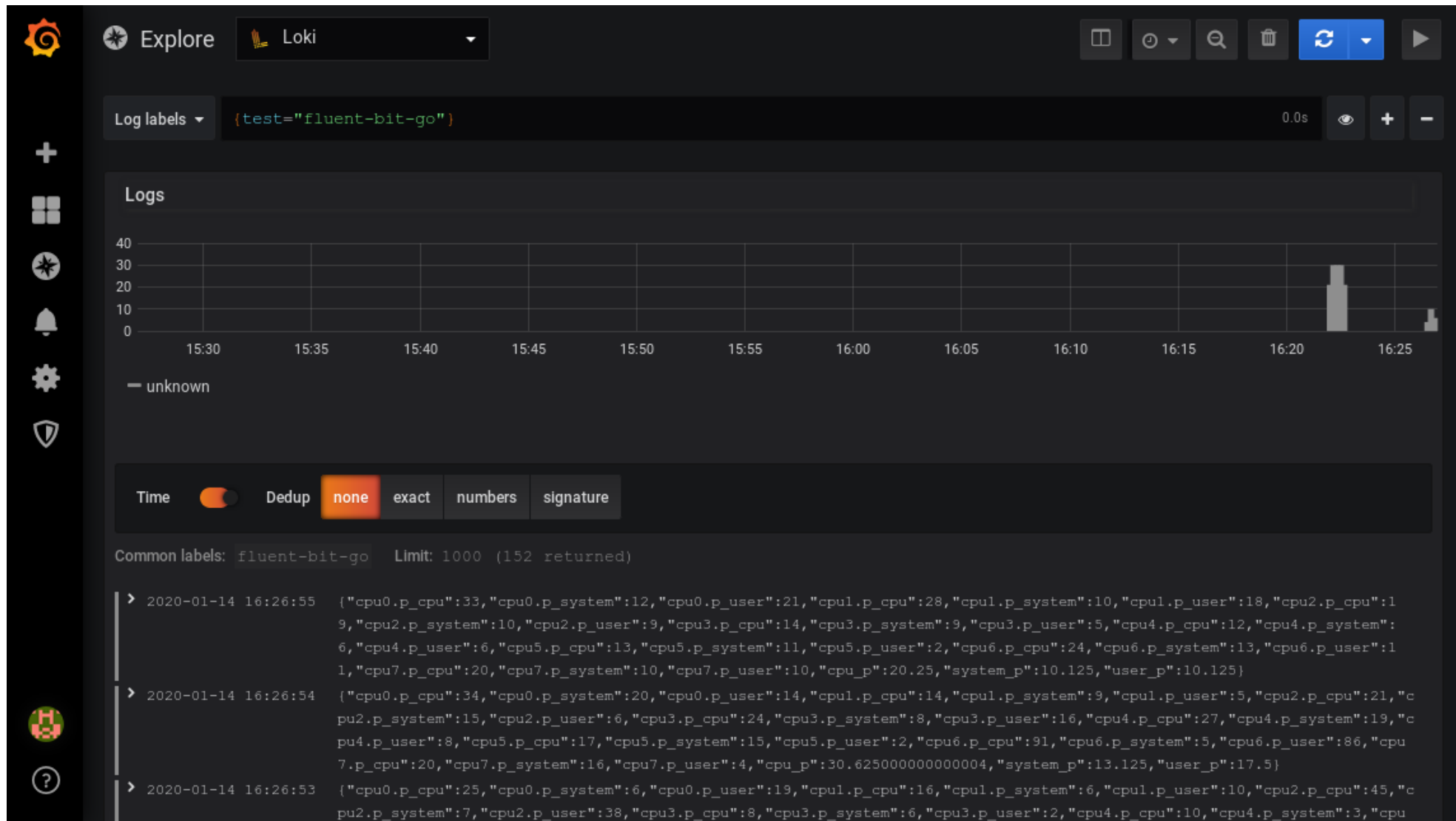
```
$ fluent-bit -e /path/to/loki.so -c /path/to/fluent.conf
```

または

```
$ cp loki.so /usr/lib/x86_64-linux-gnu/  
$ fluent-bit -c /path/to/fluent.conf
```

Grafana/Lokiのfluent-bitプラグイン

- 少し待つと、fluent-bitで取得したCPU使用率がLokiに投入される



フィードバックすると良いこと

- 独力ではすぐに対応できないプラットフォームに対応できた
 - Grafana/Lokiはk8sでの運用も見据えられている
 - fluent-bit-go-lokiのk8sへのデプロイのためのhelm chartサポートが入った
 - Grafana/Loki公式のfluent-bitプラグインのDockerイメージが提供されるようになった
 - ドキュメントもちゃんと書かれた

ちょっとした不便さや、便利さを
正しく開発元に伝えて昨日よりも
便利にしていきましょう！