

## Gitでネットワークオペレーションを進める話

2019/01/24

XFLAG STUDIO

Toshiya MABUCHI



# About

## ● 馬淵 俊弥

- 株式会社ミクシィ 開発本部
- <https://twitter.com/raibo>

## ● ふだんの仕事

- ネットワーク運用自動化(高度化)に向けたコード整備
- DPDK等でネットワークデータプレーン開発
- ネットワークまわりのコントロールプレーン/API開発



課題1:

オペレーションの時間短縮

# なぜ時間短縮が必要か

## ▶ 大前提

定期メンテ/障害時メンテ短縮し、  
ユーザへのサービス提供時間をより長く/安定させることが目的

## ▶ 定期メンテナンス時間を短く

▶ ネットワークとサーバーを同時に実施

- **サーバー** : デプロイ高度化に伴い早く/正確に
- **ネットワーク** : 運用技術向上はあるが、サーバには及ばない  
…オーバヘッドになる可能性

課題2:

オペレーションの正確さ

# オペレーションの正確さ

## ▶ 各機器の状態変化の多い作業 (e.g. ルータバージョンアップ等)

- 迂回ごとに状態確認をする  
経路差分の確認を入念に

だけど...

- ▶ 網内のNext-hop変動もある程度は確認すべきでは？
- ▶ でも現実的に手で全機器diffとるとか難しくない？

**対象機器以外で何かが起こるのがネットワーク  
他機器の状態変化もしっかり観察しないと事故るかも**

# 問題点

## ▶ 作業時間短縮

- 確認作業が大きなオーバーヘッド

## ▶ 作業の正確さ

- 確認事項が多いといつか見逃す
- 全機器の状態変化を逐一を見るのは厳しい

確認 ≡ 期待する差分が(ある)|(ない)

差分...?





# Gitを利用したネットワーク運用

## ▶ そもそも自社で既にGit(hub)を活用している

- 参考: githubでネットワーク機器のコンフィグを管理したらこんな感じ  
<https://qiita.com/junpei-yoshino/items/bf223677f332d3e833c4>

## ▶ masterで機器のConfigを管理

- 作業用ブランチを作成してPR
- PR内で変更点をレビューしてOKなら作業
- 相談事はIssueで進行

# メリット/デメリット

## ▶ よいところ

- ▶ 何を変更したか **「差分」**が見れる
- ▶ レビューでどういうことを指摘したか **「経緯」**が見れる
- ▶ 作業が完了/途中かの **「状態」**が見れる

## ▶ デメリット(?)

- ▶ メンバー全員がGitの使い方ができないといけない

作業中にも  
使えるのでは？

# Gitでオペレーション

## ▶ 作業準備以外でもGithubを活用する

- コマンド差分を逐一保存
- 差分が出たらそこが明確に分かる
- Webベースで変更差分が見やすい
- 対象機器の周辺ルータも状況も管理しやすい
- 操作用APIが既に十分完備されている

取り組み例：

# 経路差分を確認する Slack&Github Bot

# Gitを利用したネットワーク運用

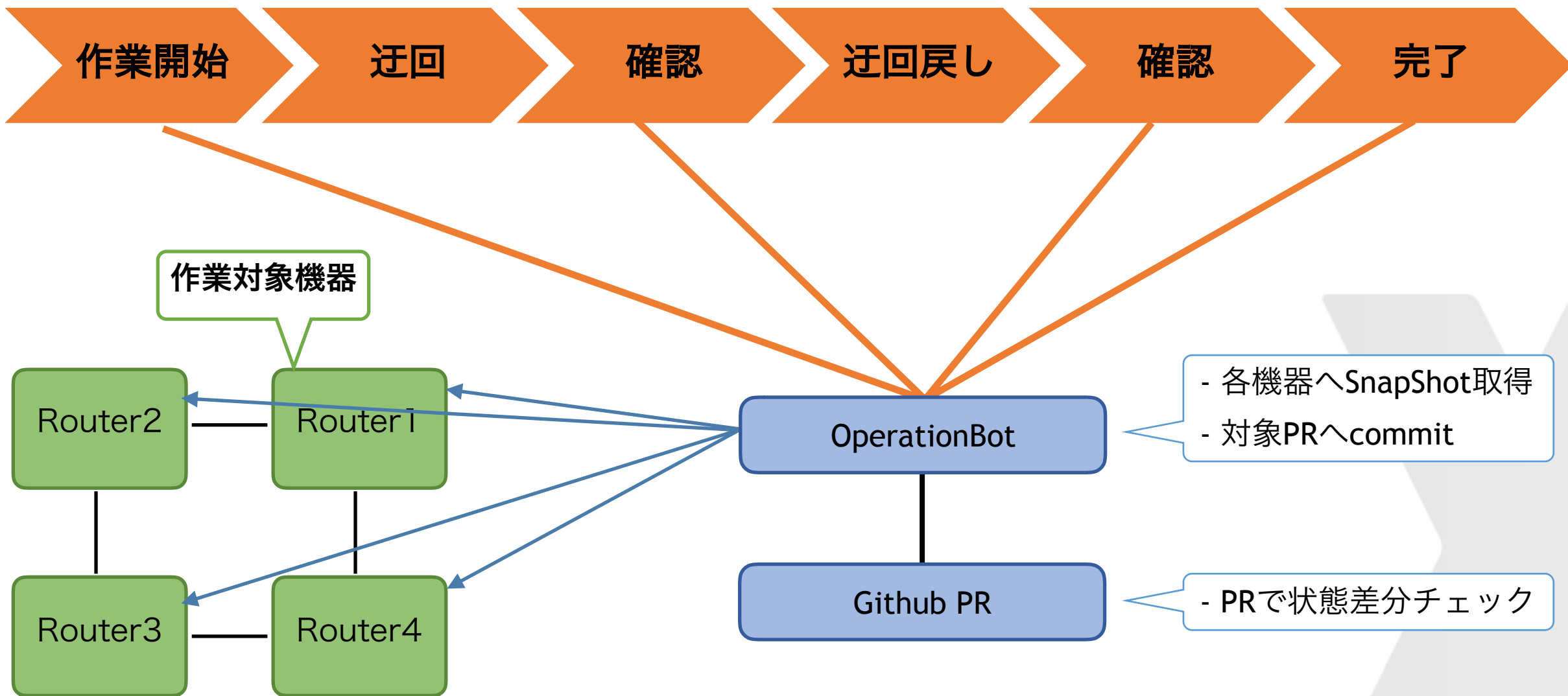
## ▶ Slackで作業開始を宣言

- ▶ 「開始時点の経路スナップショット」を全ルータから取得
  - ▶ 起点「0番」

## ▶ 作業の進行都度 「diff route」 命令を実行 経路のスナップショットを取得しcommit/push

- ▶ 番号も1,2,3番…と進んでいく
- ▶ diffがある場合、スナップショットの差分をGithubで指摘

# 経路差分を確認するSlack&Github Bot





# 作業開始



**toshiya.mabuchi** 21:35

netbot startpr 3123



**netbot** アプリ 21:35

開始準備中...ちょっと待っててね



**LAB-core2** をPR#3123のため **toshiya.mabuchi** がLockしました



経路保存を開始します... (mode = **rapid**)



事前状態保存完了. Checkpoint: **000** として記録しました



準備完了. 作業を初めて下さい!

# 作業途中で経路確認



**toshiya.mabuchi** 21:31

netbot diff route all



**netbot** アプリ 21:31

現在の経路情報を取得してきます。ちょっと待っててね

スキャン対象 : all



現在の経路情報を Checkpoint: **001** として記録しました。

CheckPoint: **000** の差分を表示します



差分があります。

[https://github.com/\[redacted\]\\_compare/8c377196799a584d4e687c778f4ed65cb4e5b988...f5ca64f25adc57c](https://github.com/[redacted]_compare/8c377196799a584d4e687c778f4ed65cb4e5b988...f5ca64f25adc57c)

# 差分があればGithub上で指摘

14 LAB-core2_route_dump.txt			
@@ -64,7 +64,7 @@ __juniper_private4__.	inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 h		
64 192.168.1.2/32	*[Local/0]	64 192.168.1.2/32	*[Local/0]
65	Local via em2.32768	65	Local via em2.32768
66		66	
67 - gre-test.inet.0: 227 destinations, 227 routes (225 active, 0 holddown, 2 hidden)		67 + gre-test.inet.0: 221 destinations, 221 routes (219 active, 0 holddown, 2 hidden)	
68 + = Active Route, - = Last Active, * = Both		68 + = Active Route, - = Last Active, * = Both	
69		69	
70 10.0.0.0/16	[BGP ], localpref 100	70 10.0.0.0/16	[BGP ], localpref 100
@@ -77,10 +77,6 @@ gre-test.inet.0: 227 destinations, 227 routes (225 active, 0 holddown, 2 hidden)			
77	> via gr-0/0/0.0	77	> via gr-0/0/0.0
78 192.168.128.1/32	*[Local/0]	78 192.168.128.1/32	*[Local/0]
79	Local via gr-0/0/0.0	79	Local via gr-0/0/0.0
80 - 192.168.128.8/30	*[Direct/0]		
81 -	> via gr-0/0/0.2		
82 - 192.168.128.9/32	*[Local/0]		
83 -	Local via gr-0/0/0.2		
84 192.168.128.12/30	*[Direct/0]	80 192.168.128.12/30	*[Direct/0]
85	> via gr-0/0/0.3	81	> via gr-0/0/0.3
86 192.168.128.13/32	*[Local/0]	82 192.168.128.13/32	*[Local/0]
@@ -109,10 +105,6 @@ gre-test.inet.0: 227 destinations, 227 routes (225 active, 0 holddown, 2 hidden)			
109	> via gr-0/0/0.9	105	> via gr-0/0/0.9
110 192.168.128.37/32	*[Local/0]	106 192.168.128.37/32	*[Local/0]
111	Local via gr-0/0/0.9	107	Local via gr-0/0/0.9
112 - 192.168.128.40/30	*[Direct/0]		
113 -	> via gr-0/0/0.10		
114 - 192.168.128.41/32	*[Local/0]		
115 -	Local via gr-0/0/0.10		
116 192.168.128.44/30	*[Direct/0]	108 192.168.128.44/30	*[Direct/0]
117	> via gr-0/0/0.11	109	> via gr-0/0/0.11
118 192.168.128.45/32	*[Local/0]	110 192.168.128.45/32	*[Local/0]
@@ -189,10 +181,6 @@ gre-test.inet.0: 227 destinations, 227 routes (225 active, 0 holddown, 2 hidden)			
189	> via gr-0/0/0.29	181	> via gr-0/0/0.29
190 192.168.128.117/32	*[Local/0]	182 192.168.128.117/32	*[Local/0]
191	Local via gr-0/0/0.29	183	Local via gr-0/0/0.29
192 - 192.168.128.120/30	*[Direct/0]		
193 -	> via gr-0/0/0.30		
194 - 192.168.128.121/32	*[Local/0]		
195 -	Local via gr-0/0/0.30		
196 192.168.128.124/30	*[Direct/0]	184 192.168.128.124/30	*[Direct/0]
197	> via gr-0/0/0.31	185	> via gr-0/0/0.31
198 192.168.128.125/32	*[Local/0]	186 192.168.128.125/32	*[Local/0]
@@			

# まとめ

- ▶ 差分を早く / 確認点がある場合のみ出すことで作業簡略化
  - ▶ 作業時間の短縮
- ▶ 見れていなかった経路差分が見れるように / 差分が明確に
  - 正確性の向上

# 課題

## ▶ CLIコマンド

- (差分表示においては)Readabilityが少し低め
- Indent処理済みJSONブロックとかなら見やすい
  - …が、NETCONF -> JSONパースが時間かかりすぎる
  - 遅いのはオペレーターのストレス

## ▶ ある機器で実行したらbfdが遅延してIGPが落ちた😰😰😰😰😰

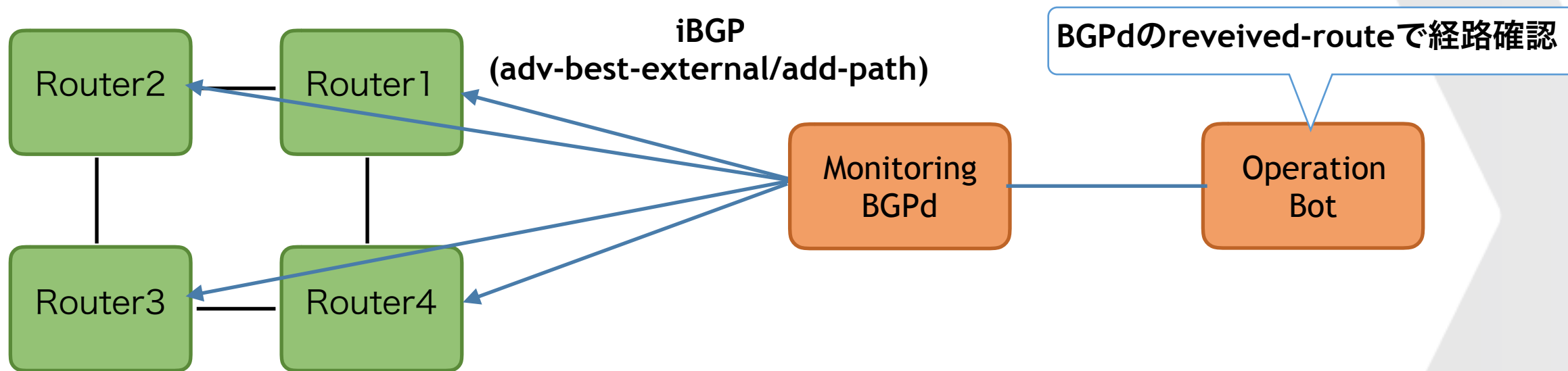
- show routeの負荷…?
- Sensitiveな場所の確認は機器外でやるべきかも

# NextStep

直接機器でやらず、各機器でiBGPを接続

Advertise best externalを使ってBest以外の変化も取得

...する予定！



**Thank you!**

