

JANOG43

ストレスフリーなオペレーション環境を目指して

2019.1.16

株式会社ミクシィ

山田千紗



XFLAG

ケタハズレな冒険を。

本日の流れ

- 01 自己紹介
- 02 背景
- 03 オペレーションフロー
- 04 まとめ

自己紹介

山田千紗（やまだ ちさや）

所属：株式会社ミクシィ 開発本部 基盤開発G

出身：山口県下関市

2015年 某ISP子会社へ新卒で入社

→BtoB, L2運用

Ringプロトコルで苦しむ

2018年 株式会社ミクシィへ入社

→BtoC, 自社サービスBB運用

BGPを勉強中

- NW業界4年目です。
- JANOG参加は2回目、登壇は初めて



背景

背景

①管理対象の事情

- ほとんどの機器が同じメーカー
- 機器の台数自体はさほど多くない（数十台規模）

②システムの事情

- 構成変更が頻繁に発生する
- スピード感のある対応が必要

③チーム体制の事情

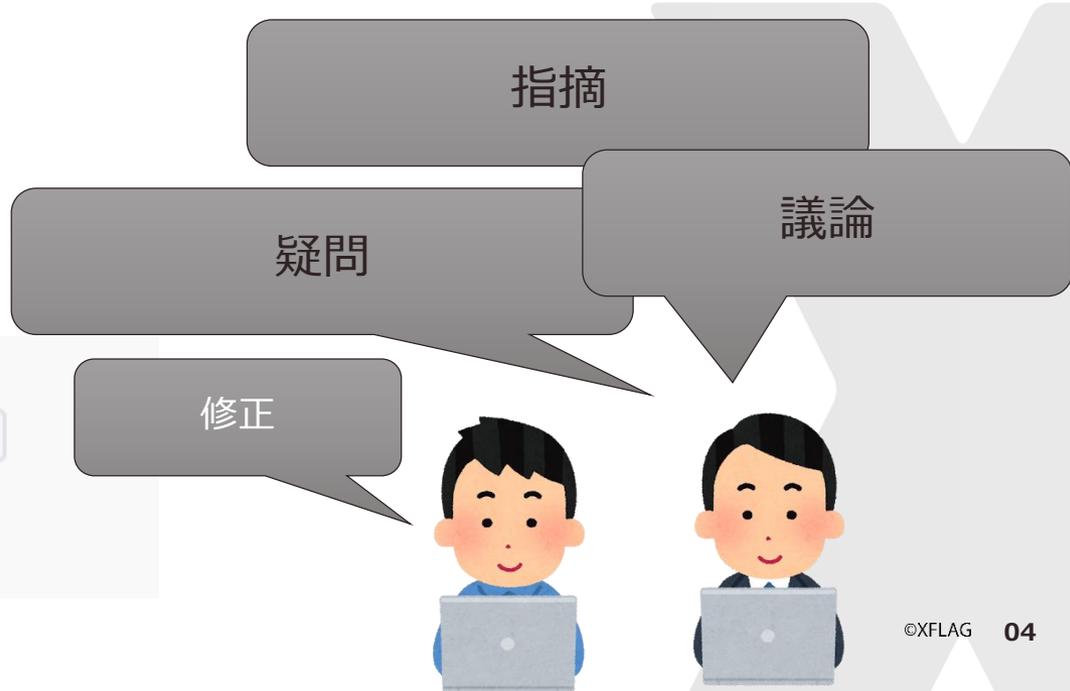
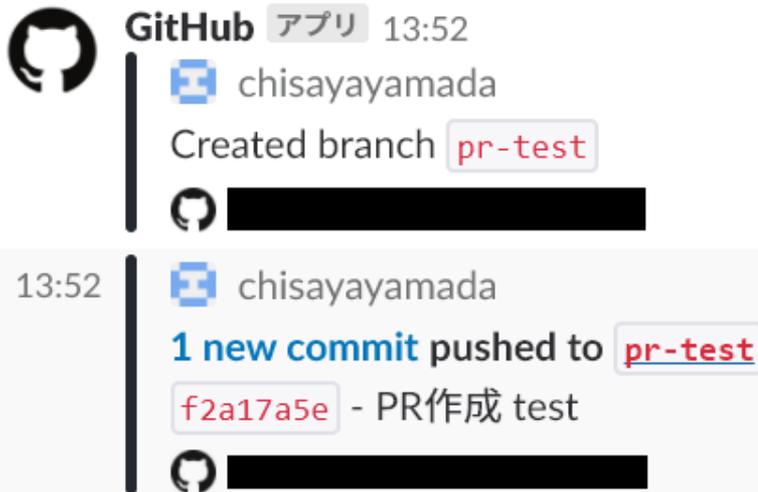
- リモートワークのメンバーがいる
- DCでの物理作業とオペレーションを一人で担当することもある
⇒ 顔を合わせてのオペレーションが困難

フロー

作業準備

実機のconfig差分を作成し、PRを上げる

- 全機器のconfigはGitHub上で管理
- commitやcommentはBOTがチームのチャンネルへ通知
 - PR内での議論の内容もチーム全員へ共有される
 - PRの内容について広い意見が得られ、オペレーションの品質が向上する



作業準備

実機のconfig差分を作成し、PRを上げる デメリット：configは手打ち

⇒ 構文ミスやtypoが生じる

PR作成時にconfigのsyntaxを自動でcheckするツールを設置

junoser (<https://github.com/codeout/junoser>)

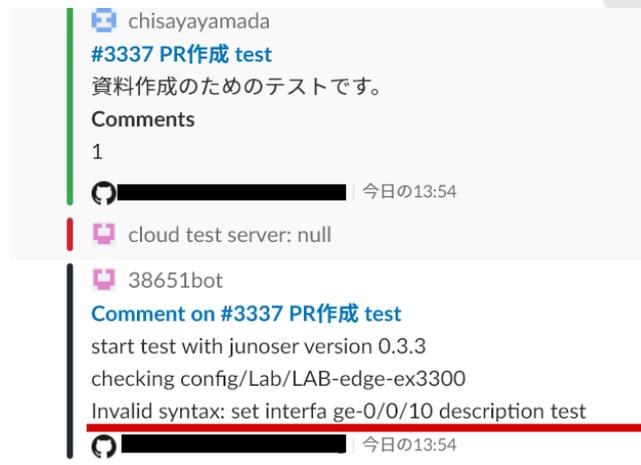
```
136 136 set interfaces ge-0/0/10 disable
137 + set interfa ge-0/0/10 description test
```

構文ミス

【GitHub】



【Slack】



作業中

screenコマンドで画面共有

作業中のコミュニケーションはslack上

- リモートから4eyesが可能
- 同じ画面を見るため、認識の共有がしやすい

【Slack】

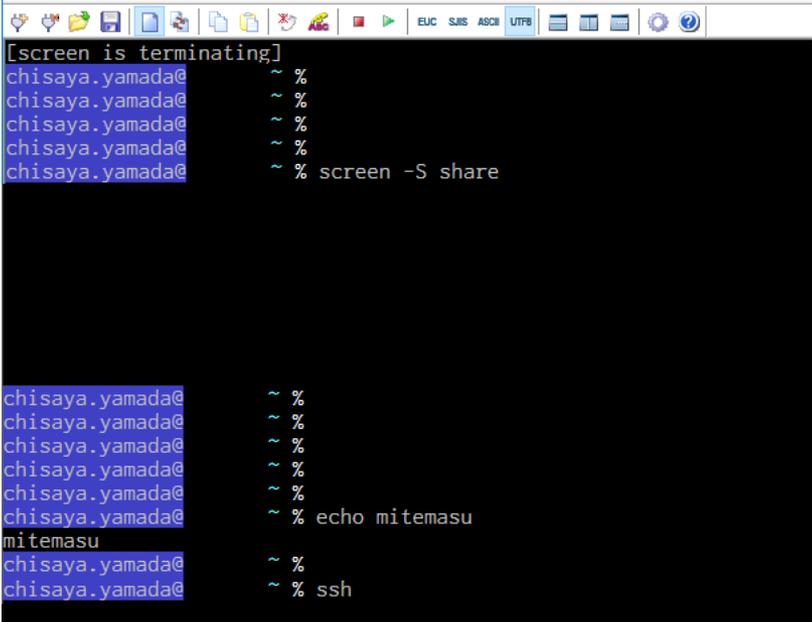
chisaya.yamada 16:02
share開きました

██████████ 16:02
echo しました

chisaya.yamada 16:02
ありがとうございます 確認しました

██████████ 16:03
██████████ host ok

【Terminal】



```
[screen is terminating]
chisaya.yamada@ ~ %
chisaya.yamada@ ~ %
chisaya.yamada@ ~ %
chisaya.yamada@ ~ %
chisaya.yamada@ ~ % screen -S share

chisaya.yamada@ ~ %
chisaya.yamada@ ~ % echo mitemasu
mitemasu
chisaya.yamada@ ~ %
chisaya.yamada@ ~ % ssh
```

作業中

状態確認

- 経路情報やBGP neighborの状態チェックは Slack BOTで実施
- GitHub上で差分確認

```
@@ -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]
65 Local via em2.32768
66
67 - gre-test.inet.0: 227 destinations, 227 routes (225 active, 0 holddown, 2 hidden)
68 + = Active Route, - = Last Active, * = Both
69
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
78 192.168.128.1/32 *[Local/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-0/0/0.0
64 192.168.1.2/32 *[Local/0]
65 Local via em2.32768
66
67 + gre-test.inet.0: 221 destinations, 221 routes (219 active, 0 holddown, 2 hidden)
68 + = Active Route, - = Last Active, * = Both
69
70 10.0.0.0/16 [BGP ], localpref 100
77 > via gr-0/0/0.0
78 192.168.128.1/32 *[Local/0]
79 Local via gr-0/0/0.0
```

 chisaya.yamada 18:40
netbot bgp peer TRANSIT_IPv4

 netbot アプリ 18:40
✔ TRANSIT_IPv4 受信経路数 1, 広報経路数 6

受信経路

- 5 nbr / 3 actv :

広報経路

- 5 nbr / 2 actv :
- 4 nbr / 2 actv :
- 4 nbr / 2 actv :
- 5 nbr / 2 actv :
- 5 nbr / 2 actv :
- 1 nbr / 1 actv :

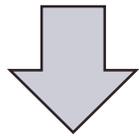
⚠ 210.138.202.240/28 はBGP接続が非冗長です。

Gitベースなオペレーション高度化

<https://www.janog.gr.jp/meeting/janog43/program/gitops>

事後作業

- スクリプトで実機configを取得
 - xloginを利用
- 再度差分をチェック
- Masterリポジトリへpush



作業完了！

まとめ

まとめ

確認が煩雑な箇所はBOTを活用

- 必ずしもshowの結果すべてを見る必要はない
- ポイントとなる情報をピックアップし、確認者の負担を軽減

screenコマンドが便利

- 確認/作業担当以外の人でも別の業務をしつつ作業を追える
 - トラブル発生時にも生のオペレーションログを追えるため情報共有が迅速

もっとイケてるやり方があるよ！

ここはこうしたほうがよくなるんじゃない？

など、コメントありましたらぜひお声がけください！



