

ネットワークコンフィグ分析ツール

# Batfish

## との付き合い方

---

2019/04/26

JANOG43.5 Interim Meeting



株式会社 エーピーコミュニケーションズ  
横地 晃 (@akira6592)

# 自己紹介

|    |                    |            |                                                                                     |
|----|--------------------|------------|-------------------------------------------------------------------------------------|
| 名前 | 横地 晃               | @akira6592 |  |
| 所属 | 株式会社エーピーコミュニケーションズ |            |                                                                                     |
| 業務 | ネットワーク自動化関連の技術開発   |            |                                                                                     |

JANOG43: 自動化の行き着く先は?、ハッカソンスタッフ

JANOG42: Ansible ネットワーク自動化チュートリアル

JANOG41: ネットワーク運用自動化BoF

ブログ : てくなべ <https://tekunabe.hatenablog.jp>

Software Design 2018年12月号 Ansible 特集

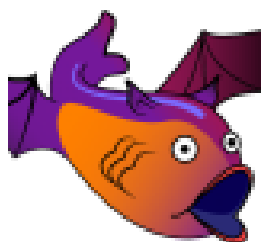


# はじめに

---

Batfish は、ネットワーク機器のコンフィグのさまざまな分析、検証ができるオープンソースのツールです。

Batfish の概要と、ツール調査の中で見えてきた使いどころなど、Batfish との付き合い方をお伝えします。



JANOG ハッカソンのネタにも

# 本日の内容

---

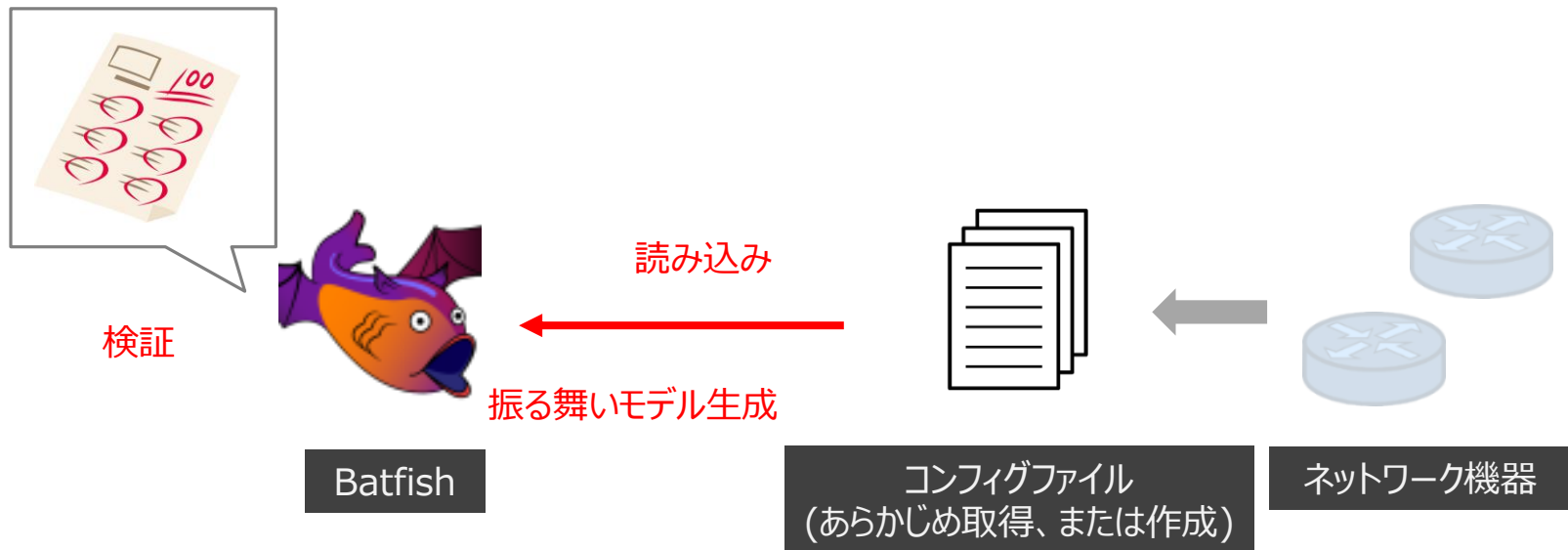
- Batfish とは
- サンプル
- 付き合いかた
- まとめ

# Batfish とは

---

# Batfish とは

- ネットワーク機器のコンフィグの様々な分析、検証ができるオープンソースのツール
- コンフィグファイルを読み込んで処理するため、実機接続は不要



# 対応プラットフォーム

---

- Arista
- Aruba
- AWS
  - (VPCs, Network ACLs, VPN GW, NAT GW, Internet GW, Security Groups, etc…)
- Cisco
  - (All Cisco NX-OS, IOS, IOS-XE, IOS-XR and ASA devices)
- Dell Force10
- F5 BIG-IP
- Foundry
- iptables (on hosts)
- Juniper
  - (All JunOS platforms: MX, EX, QFX, SRX, T-series, PTX)
- MRV
- Palo Alto Networks
- Quagga / FRR
- Quanta
- VyOS

引用元: Supported Network Device and Operating System List

<https://github.com/batfish/batfish#supported-network-device-and-operating-system-list>

# できること（一例）

---

## ● 経路確認

- traceroute が想定通りか
- ループ検出

## ● セキュリティ確認

- 指定の条件の通信が想定通り許可/拒否されるか

## ● コンフィグ妥当性確認（凡ミス検出）

- インタフェースのIPアドレスが重複していないか
- NTP/DNSサーバーの値が想定通りか
- 定義済みだけど未適用のフィルタはないか

【サンプル】

# 到達性差分 の検出

---

# 準備：コンフィグ読み込み

変更前コンフィグ読み込み



変更後コンフィグ読み込み

(RT2のスタティックルート削除)

```
hostname rt1
!
interface GigabitEthernet0/0
 ip address 172.16.1.254 255.255.255.0
!
interface GigabitEthernet1/0
 ip address 10.0.0.254 255.255.255.0
!
ip route 172.16.2.0 255.255.255.0 10.0.0.253
```

```
hostname rt2
!
interface GigabitEthernet0/0
 ip address 172.16.2.254 255.255.255.0
!
interface GigabitEthernet1/0
 ip address 10.0.0.253 255.255.255.0
!
ip route 172.16.1.0 255.255.255.0 10.0.0.254
```

172.16.1.0/24

host  
101

rt1

rt2

host  
201

172.16.2.0/24

```
hostname rt1
!
interface GigabitEthernet0/0
 ip address 172.16.1.254 255.255.255.
!
interface GigabitEthernet1/0
 ip address 10.0.0.254 255.255.255.0
!
ip route 172.16.2.0 255.255.255.0 10.0.0.253
```

```
hostname rt2
!
interface GigabitEthernet0/0
 ip address 172.16.2.254 255.255.255.0
!
interface GigabitEthernet1/0
 ip address 10.0.0.253 255.255.255.0
!
```

```
ip route 172.16.1.0 255.255.255.0 10.0.0.254
```

ルート削除

# 到達性差分の検出

differentialReachability により、2つのコンフィグのセット  
(スナップショット) 間の到達性の差分を検出

```
answer_diff = bfq.differentialReachability(  
    pathConstraints=PathConstraints(startLocation="@enter(/.+/)"),  
    headers=HeaderConstraints(dstIps="/.+/")  
)  
answer =  
    snapshot=BASE_NAME,  
    reference_snapshot=CHANGED1_NAME)  
show(answer_diff.frame())
```

ルート削除により、  
到達性がなくなる通信を3つ検出

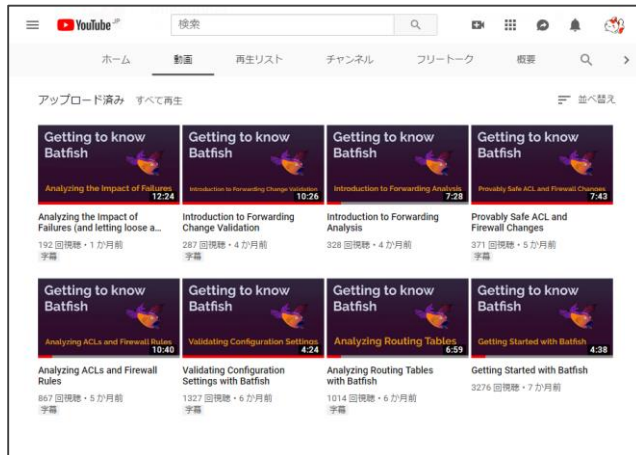
| 変更前                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 変更後                 |                                                                                                                                                                                                                    |   |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| Flow                                                                                                                 | Snapshot_Traces                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Snapshot_TraceCount | Reference_TraceCount                                                                                                                                                                                               |   |
| Start Location: host201<br>interface=eth0<br>Src IP: 172.16.2.2<br>Dst IP: 172.16.1.1<br>IP Protocol: ICMP           | ACCEPTED<br>1. node: host201<br>RECEIVED(eth0)<br>FORWARDED(Routes: static [Network: 0.0.0.0/0,<br>Next Hop IP: 172.16.2.254])<br>TRANSMITTED(eth0)<br>2. node: rt2<br>RECEIVED(GigabitEthernet0/0)<br>FORWARDED(Routes: static [Network:<br>172.16.1.0/24, Next Hop IP: 10.0.0.254])<br>TRANSMITTED(GigabitEthernet1/0)<br>3. node: rt1<br>RECEIVED(GigabitEthernet1/0)<br>FORWARDED(Routes: connected [Network:<br>172.16.1.0/24, Next Hop IP: AUTO/NONE(-1)])<br>TRANSMITTED(GigabitEthernet0/0)<br>4. node: host101<br>RECEIVED(eth0)<br>ACCEPTED(InboundStep) | 1                   | NO_ROUTE<br>1. node: host201<br>RECEIVED(eth0)<br>FORWARDED(Routes: static<br>[Network: 0.0.0.0/0, Next Hop<br>IP: 172.16.2.254])<br>TRANSMITTED(eth0)<br>2. node: rt2<br>RECEIVED(GigabitEthernet0/0)<br>NO_ROUTE | 1 |
| 0                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                     |                                                                                                                                                                                                                    |   |
| Start Location: rt2<br>interface=GigabitEthernet0/0<br>Src IP: 172.16.2.2<br>Dst IP: 172.16.1.1<br>IP Protocol: ICMP | ACCEPTED<br>1. node: rt2<br>RECEIVED(GigabitEthernet0/0)<br>FORWARDED(Routes: static [Network:<br>172.16.1.0/24, Next Hop IP: 10.0.0.254])<br>TRANSMITTED(GigabitEthernet1/0)<br>2. node: rt1<br>RECEIVED(GigabitEthernet1/0)<br>FORWARDED(Routes: connected [Network:<br>172.16.1.0/24, Next Hop IP: AUTO/NONE(-1)])<br>TRANSMITTED(GigabitEthernet0/0)<br>3. node: host101<br>RECEIVED(eth0)<br>ACCEPTED(InboundStep)                                                                                                                                            | 1                   | NO_ROUTE<br>1. node: rt2<br>RECEIVED(GigabitEthernet0/0)<br>NO_ROUTE                                                                                                                                               | 1 |
| 1                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                     |                                                                                                                                                                                                                    |   |
| Start Location: rt2<br>interface=GigabitEthernet1/0<br>Src IP: 10.0.0.1<br>Dst IP: 172.16.1.1<br>IP Protocol: ICMP   | ACCEPTED<br>1. node: rt2<br>RECEIVED(GigabitEthernet1/0)<br>FORWARDED(Routes: static [Network:<br>172.16.1.0/24, Next Hop IP: 10.0.0.254])<br>TRANSMITTED(GigabitEthernet1/0)<br>2. node: rt1<br>RECEIVED(GigabitEthernet1/0)<br>FORWARDED(Routes: connected [Network:<br>172.16.1.0/24, Next Hop IP: AUTO/NONE(-1)])<br>TRANSMITTED(GigabitEthernet0/0)<br>3. node: host101<br>RECEIVED(eth0)<br>ACCEPTED(InboundStep)                                                                                                                                            | 1                   | NO_ROUTE<br>1. node: rt2<br>RECEIVED(GigabitEthernet1/0)<br>NO_ROUTE                                                                                                                                               | 1 |
| 2                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                     |                                                                                                                                                                                                                    |   |

# 付き合い方

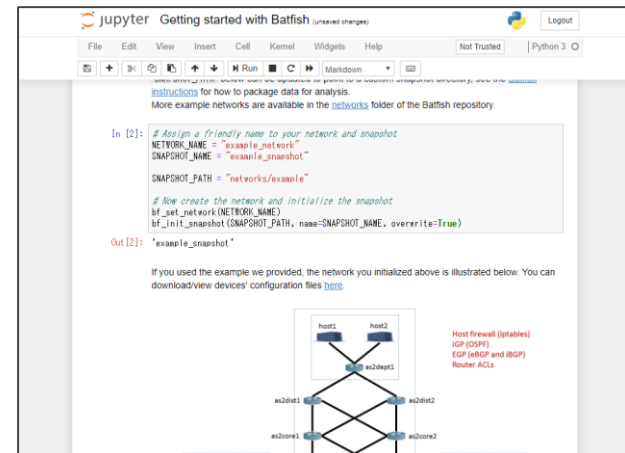
---

# どうやって学習する？

## チュートリアル動画と Jupyter Notebook が便利



[https://www.youtube.com/channel/UCA-OUW\\_3lOt9U\\_s60KvmJYA](https://www.youtube.com/channel/UCA-OUW_3lOt9U_s60KvmJYA)



batfish + pybatfish + Jupyter Notebook 一式のコンテナ

`docker run -p 8888:8888 batfish/allinone`

見るだけなら

[https://github.com/batfish/pybatfish/tree/master/jupyter\\_notebooks](https://github.com/batfish/pybatfish/tree/master/jupyter_notebooks)

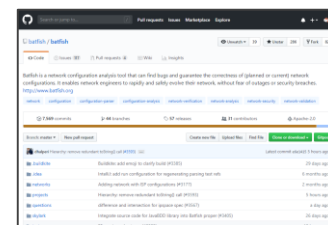
他、公式ページトップ <https://www.batfish.org/> にも参考リンクあり

# バグかなと思ったら

## GitHub や Slack で確認

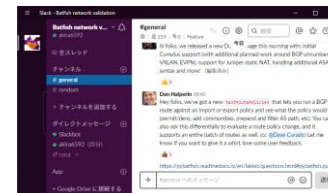
- batfish 本体、または pybatfish のissue

- <https://github.com/batfish/batfish>
- <https://github.com/batfish/pybatfish>



- batfish slack

- <https://www.batfish.org/> 上に 参加リンクあり
- レスポンスが早めで助かる



- テスト用コンフィグファイルもヒントに

- batfish/projects/batfish/src/test/resources/org/batfish/grammar/\*/testconfigs/\*
- <https://github.com/batfish/batfish/tree/master/projects/batfish/src/test/resources/org/batfish/grammar>

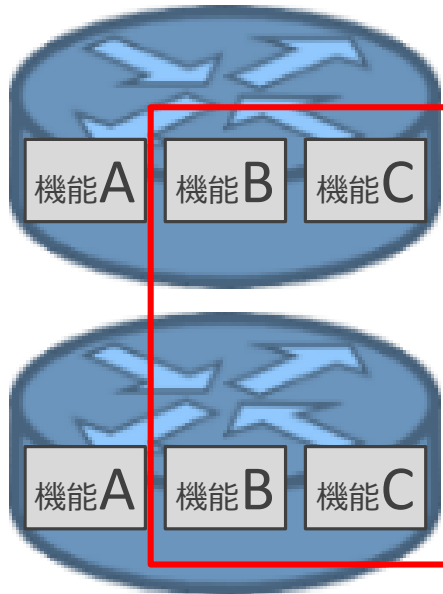
# 使いどころ

---

1. 特定の機能を
  2. 事前に
  3. 網羅的に
- 検証する

# 使いどころ1「特定の機能を」

Batfish の対応機能を調べて、  
検証対象の機能を特定する。



【調べ方】

pybatfish リファレンスの  
「Available questions」を確認する

<https://pybatfish.readthedocs.io/en/latest/questions.html>

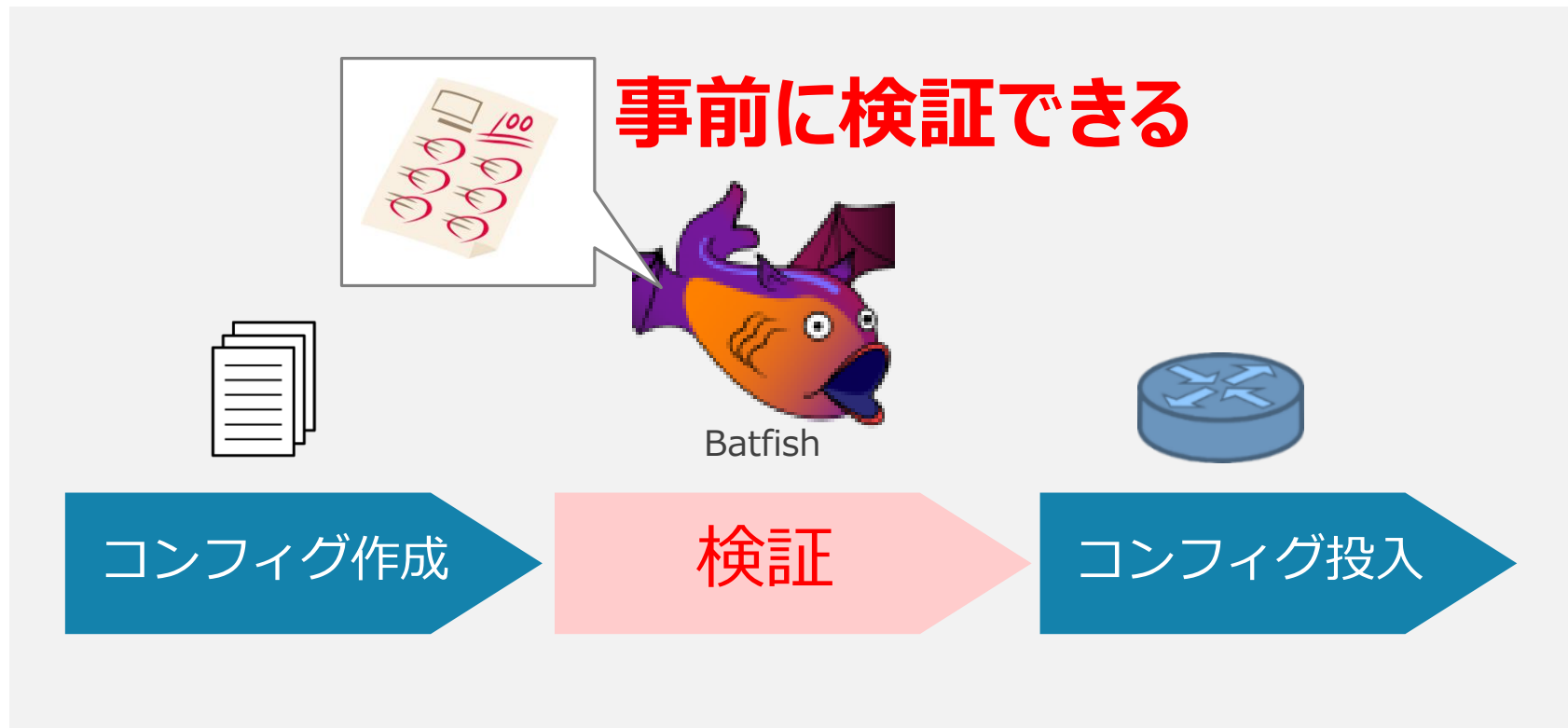
- traceroute
- routes
- searchFilters
- reachability
- detectLoops
- など

ルーティング、  
フィルター系が多い

日本語参考情報: BatfishのQuestion一覧と概要説明

[https://qiita.com/tech\\_kitara/items/be71005ad7b5091d25a4](https://qiita.com/tech_kitara/items/be71005ad7b5091d25a4)

# 使いどころ2「事前に」



実環境に手を加える前に検証、  
機器がない状態で検証

# 使いどころ3「網羅的に」

多数の

- アドレス
- 経路
- インターフェース
- ノード、など

を**網羅的に**  
いっぺんに検証

【例】

1. 複数の経路で接続されているノード間で、特定のノードを経由する通信を検出  
実装例: [Introduction to Forwarding Change Validation](#)  
> Step 2: Ensure that no traffic is routed through core1
2. コンフィグ変更前後で、到達性の有無が変わった通信を検出  
実装例: [Introduction to Forwarding Change Validation](#)  
> Step 3: Outside-to-host traffic is unaffected.
3. 全ノード、全インターフェース内で、重複したIPアドレスがないことを検証  
実装例: [重複したインターフェースのIPアドレスを検出する \(IOS/Junos混在編\)](#)

# できないこと

---

- ネットワークOS、バージョン固有の仕様、不具合の検出
- など

まとめ

---

# まとめ

---

- Batfish は、ネットワーク機器の  
コンフィグのさまざまな分析、検証が  
できるオープンソースのツール
- 特定の機能を、事前に、網羅的に検証  
するのに向いている

# 【付録】 参考資料

---

- 公式ドキュメント

- トップ

- <https://www.batfish.org/>

- チュートリアル動画

- [https://www.youtube.com/channel/UCA-OUW\\_3IOt9U\\_s60KvmJYA](https://www.youtube.com/channel/UCA-OUW_3IOt9U_s60KvmJYA)

- 日本語情報

- Batfishというconfigテストツールの可能性 (JANOG43)

- <https://www.janog.gr.jp/meeting/janog43/program/lt1>

- Batfish Advent Calendar 2018

- <https://qiita.com/advent-calendar/2018/batfish>

- ネットワーク機器のコンフィグを読み込んでルーティングなどの様々な検証ができるツール「Batfish」の紹介

- <https://tekunabe.hatenablog.jp/entry/2018/10/25/batfish>