Yahoo! JAPAN Site Operations

ヤフーの IP Clos ネットワークの歴史と運用

ヤフー株式会社
サイトオペレーション本部
インフラ技術１部 ネットワーク開発
高橋　翔

自己紹介

- 高橋　翔 (たかはし　しょう)

- 2017: 九州工業大学 卒業
- 2017/04: ヤフー入社
  データセンタネットワーク運用
  (L2/L3 スイッチ 構築/運用)
- 2019/04: ネットワーク開発
  (IP Clos の設計/構築/運用)

アジェンダ

- IP Clos Network の導入から現在まで
- 自動化への取り組み
- 構成管理ツールの遷移
  - 独自実装
  - OpenClos
  - Apstra
  - Ansible
- 構成管理ツールの使い分け
- ネットワークの監視体制

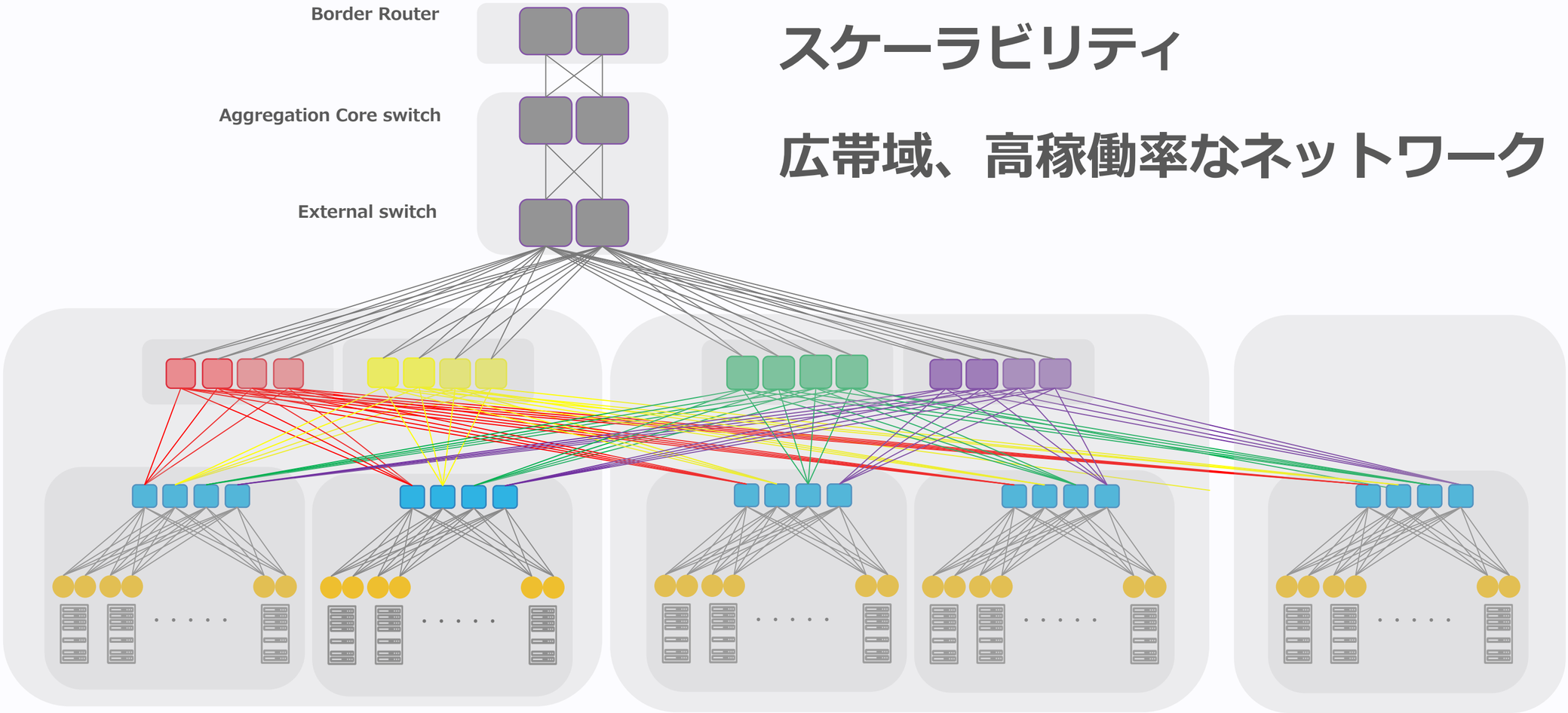
2015年ヤフー初の IP Clos Network 導入

- **JANOG38 にて Clos Network の導入事例を紹介**
  - **Hadoop 分析環境用のネットワーク**
  - **North-south → East-west traffic へ**

    https://www.janog.gr.jp/meeting/janog38/program/clos.html

ヤフーの IP Clos がどのように変化していったのか
苦労した点などをご紹介

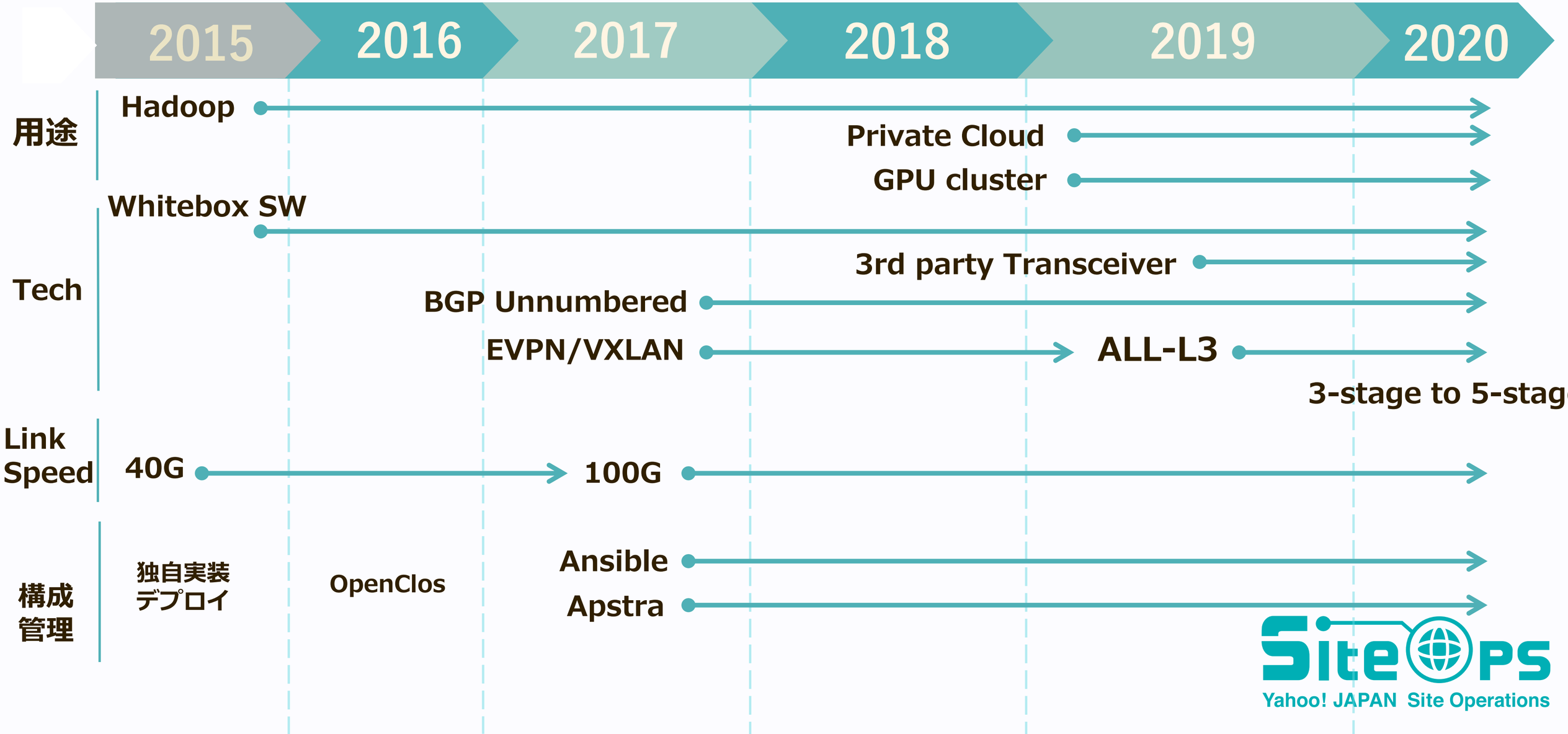
IP Clos Network とは



スケーラビリティ

広帯域、高稼働率なネットワーク

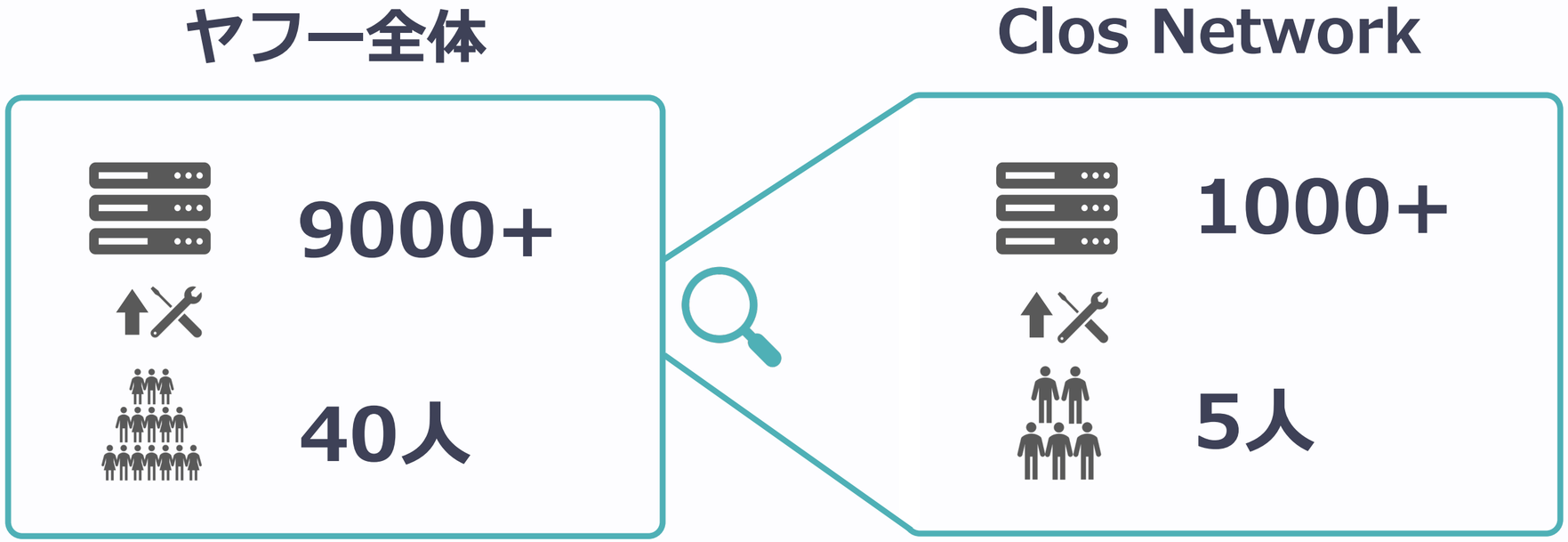
2015~現在までに数多くの UPDATE



Yahoo! JAPAN Site Operations

ヤフーのネットワーク運用体制

設計・構築・運用 を同じチームが担当



運用負荷を下げるため自動化へ取り組み

Yahoo! JAPAN Site Operations

運用負荷への取り組み方の違い

Traditional Network

- **自動化の目的: 機器のオペレーション**
- 多種多様な機器に様々な設定
  - コアスイッチ/ToRスイッチを L3/L2 接続
  - 頻繁に設定変更が必要
    - SVI, OSPF, trunk vlan...
- 内製の自動化ツールを採用
  - 多種多様な機器
  - 古すぎて API が無い

IP Clos Network

- **自動化の目的: 初期設定の低コスト化**
- 同じ種類の機器を大量に設置
  - Spine/Leaf スイッチを L3接続
  - 一度構築したら、設定を変更しない
  - 機器に合わせた設定の生成
  - 生成した設定の自動投入 (ZTP)

ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

Gen1

2015

創世期

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen2

2016

拡大期

Apstra

- running-config更新
- マルチベンダ対応
- Intent-Based 異常検知
- 操作に慣れが必要
- OSS の選択肢も欲しい

Gen3

2017~

安定拡大期

Ansible

- 広く使われているOSS
- 自由なconfig
- Git で履歴/差分確認
- マルチベンダは難しい

Gen4.0/4.1

2017~

挑戦期

SiteOPS

Yahoo! JAPAN Site Operations

Gen1: 自社製のデプロイスクリプト, 2015

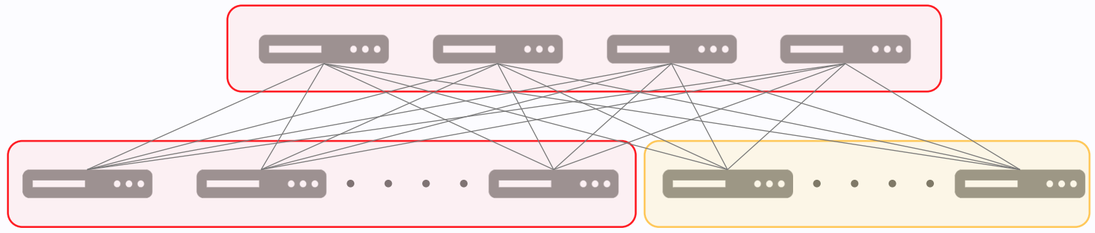
- 社内 Hadoop 基盤
- 自社IPAM + config 生成スクリプト + ZTP で構築
- NW機器
  - Spine: Arista Chassis
  - Leaf: Arista + Whitebox (Cumulus)

ヤフー社内IPAM

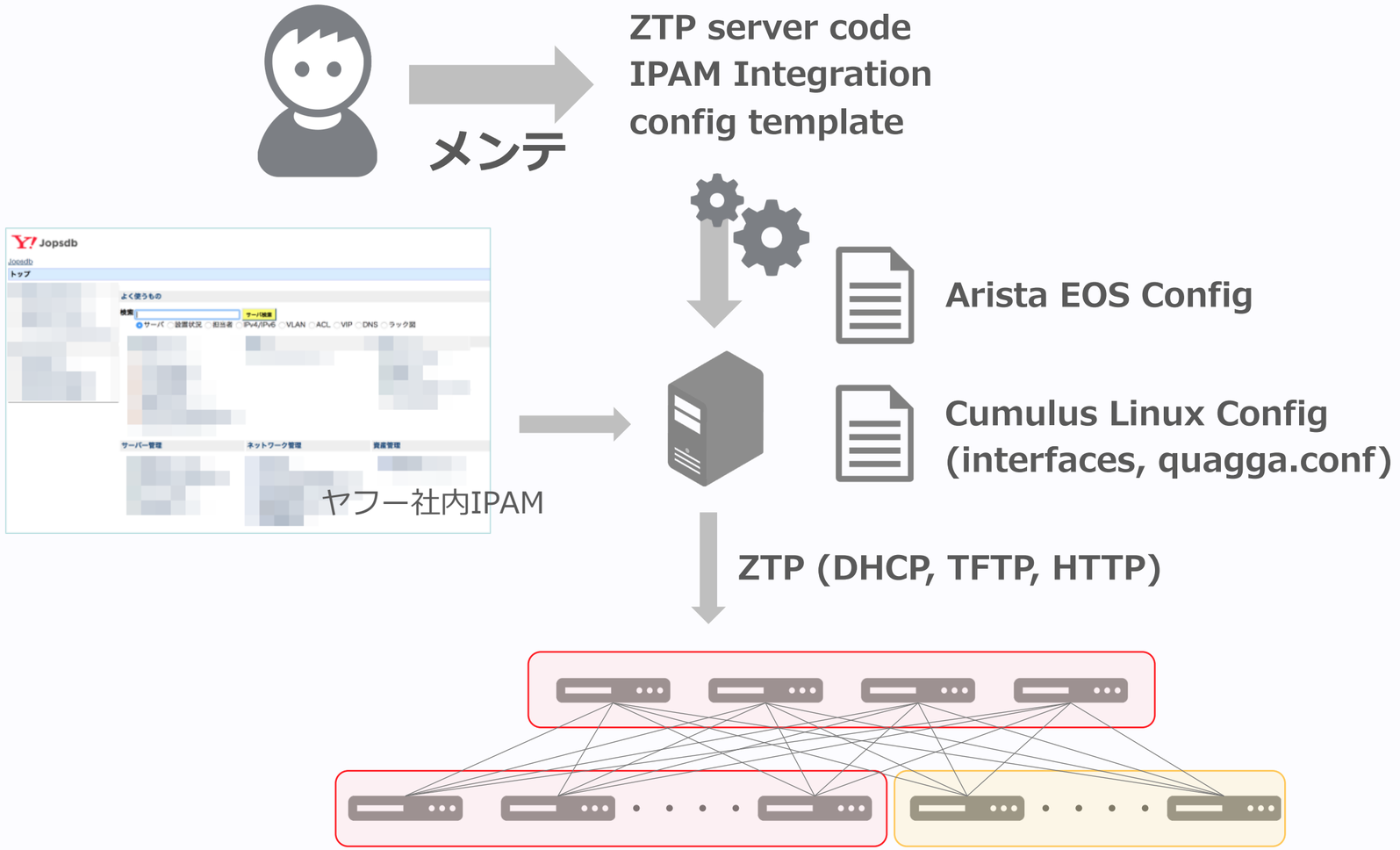
Arista EOS Config

Cumulus Linux Config
(interfaces, quagga.conf)

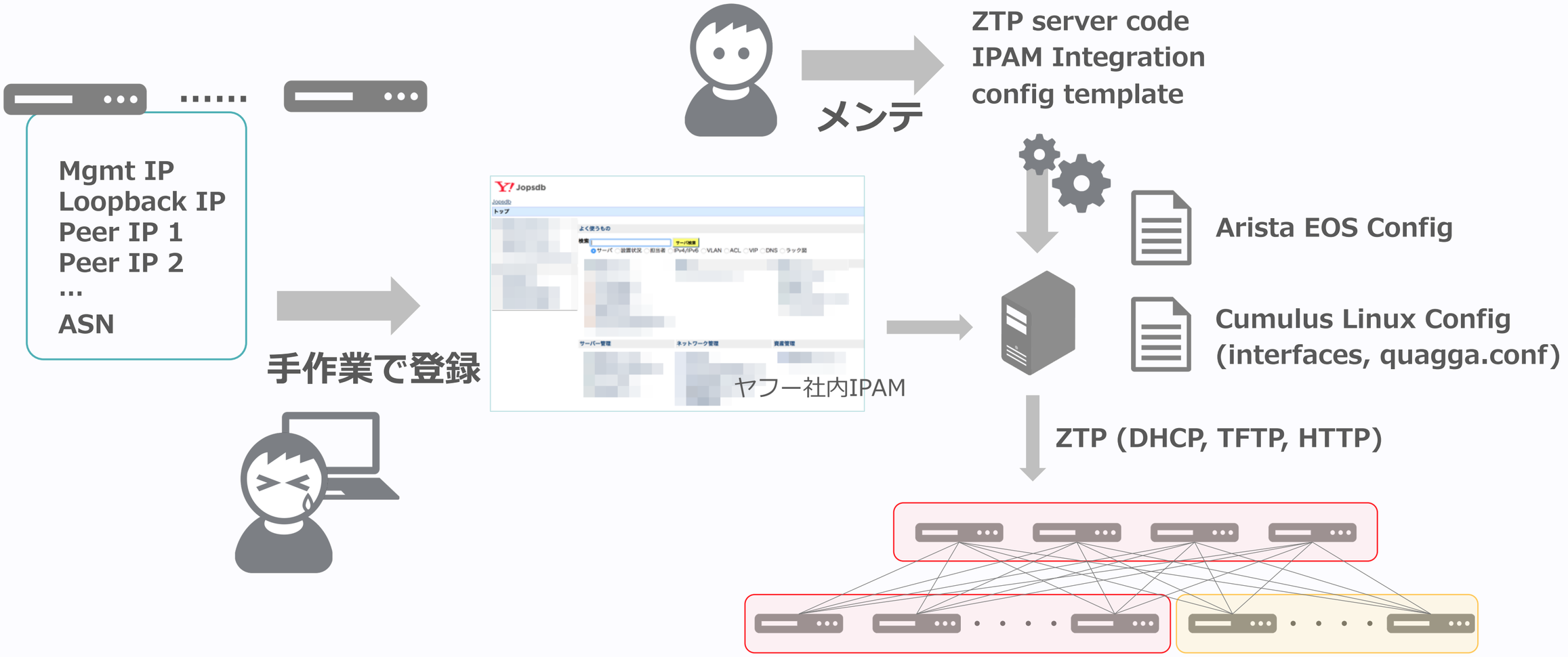
ZTP (DHCP, TFTP, HTTP)



Gen1: 内製プロダクトのメンテナンスコスト



Gen1: リソースのIPAM登録が手間



ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

Gen1

2015

創世期

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen2

2016

拡大期

Apstra

- running-config更新
- マルチベンダ対応
- Intent-Based 異常検知
- 操作に慣れが必要
- OSS の選択肢も欲しい

Gen3

2017~

安定拡大期

Ansible

- 広く使われているOSS
- 自由なconfig
- Git で履歴/差分確認
- マルチベンダは難しい

Gen4.0/4.1

2017~

挑戦期

Gen2: OpenClos, 2016

- 社内 Hadoop 基盤の第二弾
- Juniper社 OSS の OpenClos を活用して構築

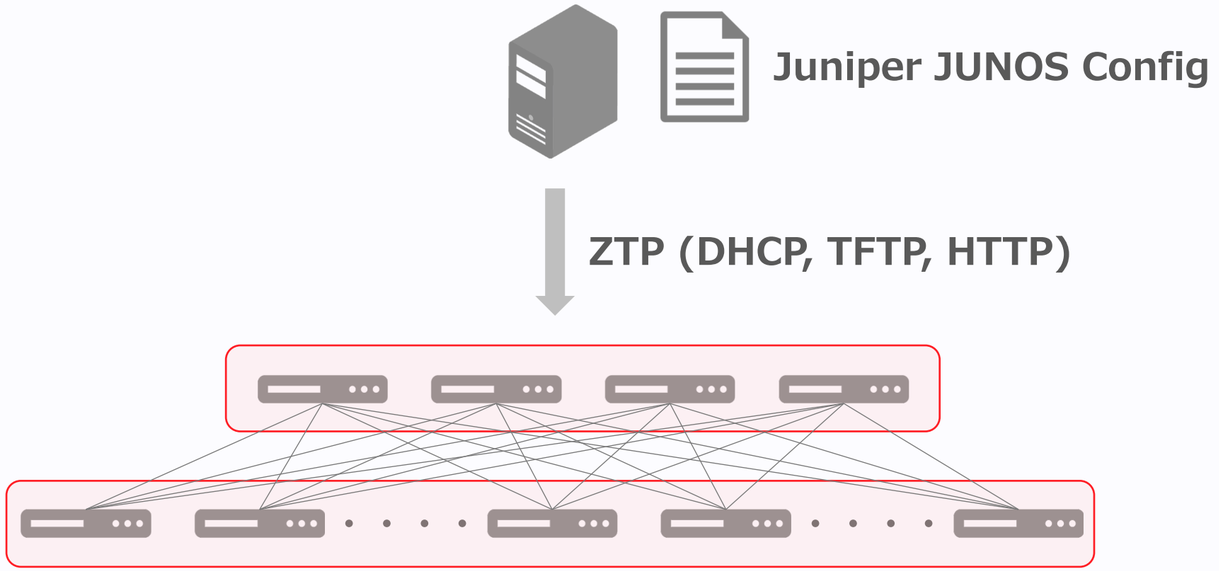
  https://github.com/Juniper/OpenClos
- NW機器
  - Spine/Leaf: Juniper

```
anotherPod:
    spineCount : 4
    # possible options for spine deviceType are qfx5100-24q-2p, qfx10002-***, qfx10008-***
    # the image file should be placed under <install dir>/jnpr/openclos/conf/ztp
    # if not placed under this dir, the file would not be accessible from http server
    # and ZTP process will be broken, these are optional, overrides global setting ztp.junosImage
    spineSettings :
        - deviceType : qfx10002-72q
          #junosImage : jinstall-qfx-5-14.1X53-D10.4-domestic-signed.tgz

    leafCount : 35
    # possible options for leafDeviceType are qfx5100-96S, qfx5100-48s-6q
    # for complete list refer to openclos.yaml
    # the image file should be placed under <install dir>/jnpr/openclos/conf/ztp
    leafSettings :
        - deviceType : qfx5100-48t-6q
          #junosImage : jinstall-qfx-5-14.1X53-D10.4-domestic-signed.tgz
        - deviceType : qfx5100-48s-6q
          #junosImage : jinstall-ex-4300-14.1X53-D10.4-domestic-signed.tgz
    # Number of uplink from leaf must be properly connected to indicate
    # the leaf device as "good". If the leaf device is not in "good" state
    # it would not go through 2-stage ZTP/configuration process.
    # Possible value is in between 2 and spineCount, inclusive both end.
    # This field is optional, default value is max(2, math.ceil(spineCount/2))
    # leafUplinkcountMustBeUp : 2

    hostOrVmCountPerLeaf : 30
    interConnectPrefix : 100.68.144.0/22
    vlanPrefix : 100.68.128.0/20
    loopbackPrefix : 100.68.148.0/24
    # either managementPrefix or (managementStartingIP, managementMask) is mandatory. Here is how it works:
    # case 1: managementPrefix : 1.2.3.7/24
    #         from cidr notation of managementPrefix we know available block is [1.2.3.0 - 1.2.3.255]
    #         from ip portion of managementPrefix we know starting ip is 1.2.3.7
    #         so the effective range is [1.2.3.7 - 1.2.3.7+spineCount+leafCount]
    # case 2: managementStartingIP : 1.2.3.7
    #         managementMask : 24
    #         from cidr notation of 'managementStartingIP/managementMask' we know availab
    #         from managementStartingIP we know starting ip is 1.2.3.7
    #         so the effective range is [1.2.3.7 - 1.2.3.7+spineCount+leafCount]
    managementPrefix : 192.168.241.56/23
```

IP Pool
ASN Pool



SiteOPS
Yahoo! JAPAN Site Operations

Gen2: OpenClos を導入して改善できたこと

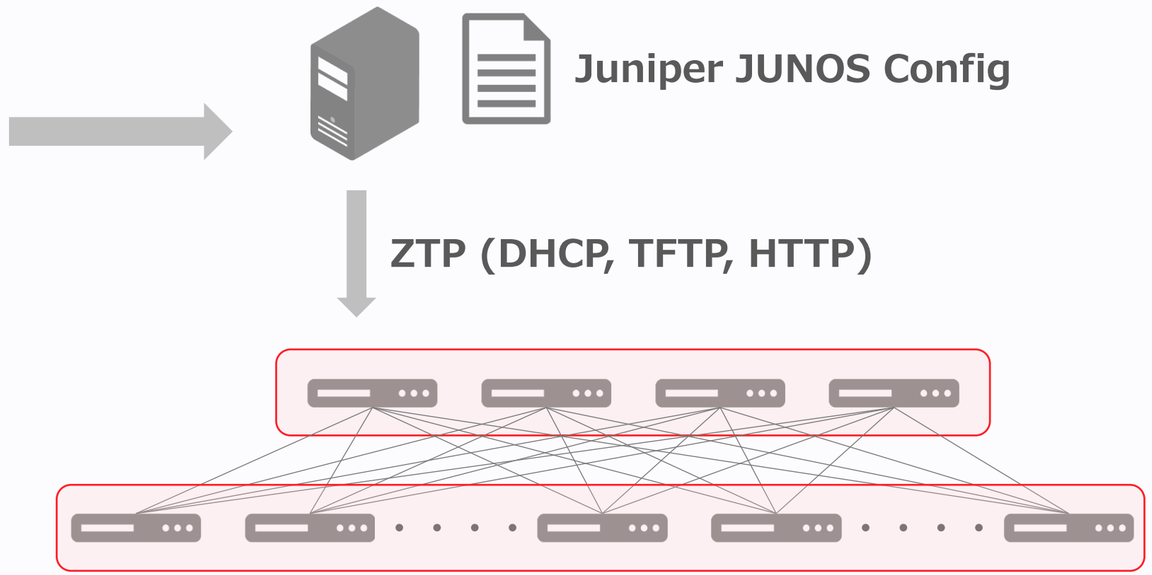
- ✓ ASN/IP を自動アサイン
- ✓ 社内 IPAM との複雑な連携が不要

```
anotherPod:
    spineCount : 4
    # possible options for spine deviceType are qfx5100-24q-2p, qfx10002-***, qfx10008-***
    # the image file should be placed under <install dir>/jnpr/openclos/conf/ztp
    # if not placed under this dir, the file would not be accessible from http server
    # and ZTP process will be broken, these are optional, overrides global setting ztp.junosImage
    spineSettings :
        - deviceType : qfx10002-72q
          #junosImage : jinstall-qfx-5-14.1X53-D10.4-domestic-signed.tgz

    leafCount : 35
    # possible options for leafDeviceType are qfx5100-96S, qfx5100-48s-6q
    # for complete list refer to openclos.yaml
    # the image file should be placed under <install dir>/jnpr/openclos/conf/ztp
    leafSettings :
        - deviceType : qfx5100-48t-6q
          #junosImage : jinstall-qfx-5-14.1X53-D10.4-domestic-signed.tgz
        - deviceType : qfx5100-48s-6q
          #junosImage : jinstall-ex-4300-14.1X53-D10.4-domestic-signed.tgz
    # Number of uplink from leaf must be properly connected to indicate
    # the leaf device as "good". If the leaf device is not in "good" state
    # it would not go through 2-stage ZTP/configuration process.
    # Possible value is in between 2 and spineCount, inclusive both end.
    # This field is optional, default value is max(2, math.celi(spineCount/2))
    # leafUplinkcountMustBeUp : 2

    hostOrVmCountPerLeaf : 30
    interConnectPrefix : 100.68.144.0/22
    vlanPrefix : 100.68.128.0/20
    loopbackPrefix : 100.68.148.0/24
    # either managementPrefix or (managementStartingIP, managementMask) is mandatory. Here is how it works:
    # case 1: managementPrefix : 1.2.3.7/24
    #         from cidr notation of managementPrefix we know available block is [1.2.3.0 - 1.2.3.255]
    #         from ip portion of managementPrefix we know starting ip is 1.2.3.7
    #         so the effective range is [1.2.3.7 - 1.2.3.7+spineCount+leafCount]
    # case 2: managementStartingIP : 1.2.3.7
    #         managementMask : 24
    #         from cidr notation of 'managementStartingIP/managementMask' we know available block is [1.2.3.0 - 1.2.3.255]
    #         from managementStartingIP we know starting ip is 1.2.3.7
    #         so the effective range is [1.2.3.7 - 1.2.3.7+spineCount+leafCount]
    managementPrefix : 192.168.241.56/23
```

IP Pool
ASN Pool



SiteOPS
Yahoo! JAPAN Site Operations

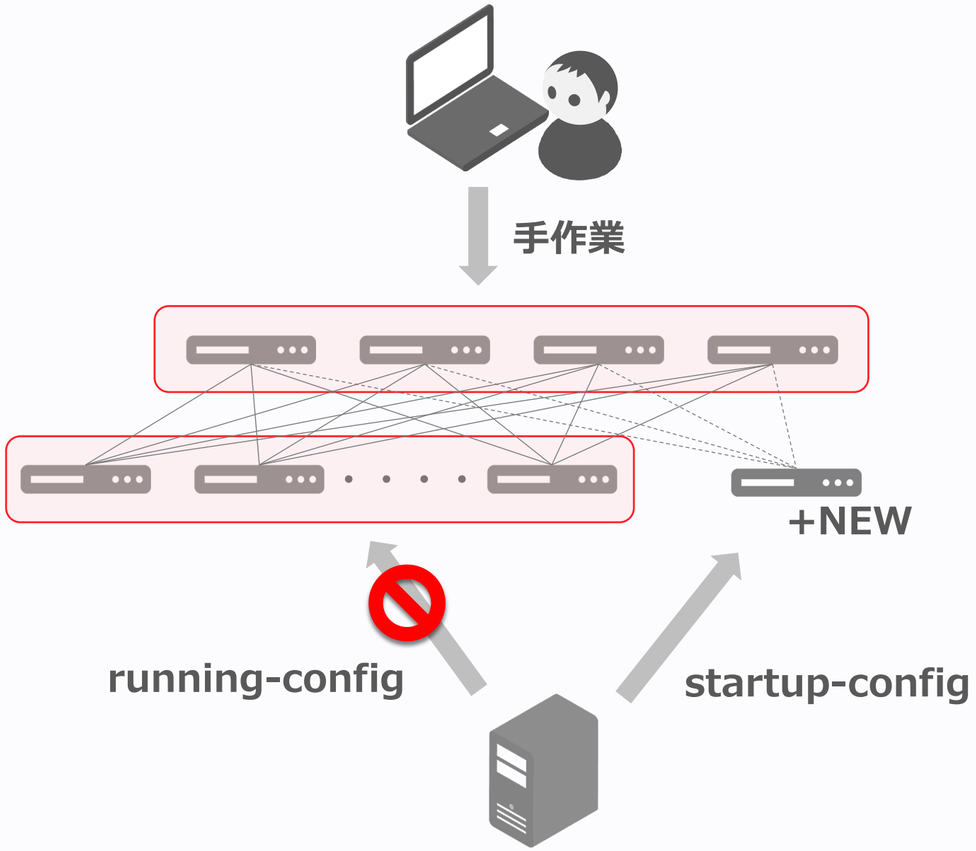
Gen2: OpenClos での課題

サービスイン後の手作業が多かった

- startup-config は OpenClos で生成できる
- running-config の update ができなかった
  - 新しい Leaf 追加するとき

    稼働中 Spine への Leaf 接続は手作業
  - メンテナンス作業

- 運用途中で他ベンダースイッチの接続が必要に
  - 初期設定に対応できなかった



ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen1
2015
創世期

Gen2
2016
拡大期

構築の自動化が主目的

Clos Network が
一度に、大規模に構築するもの

スケールアウト時のコストが高い

汎用的に使えて
スケールアウトしやすいツール

ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

Gen1

2015

創世期

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen2

2016

拡大期

Apstra

- running-config更新
- マルチベンダ対応
- Intent-Based 異常検知
- 操作に慣れが必要
- OSS の選択肢も欲しい

Gen3

2017~

安定拡大期

Ansible

- 広く使われているOSS
- 自由なconfig
- Git で履歴/差分確認
- マルチベンダは難しい

Gen4.0/4.1

2017~

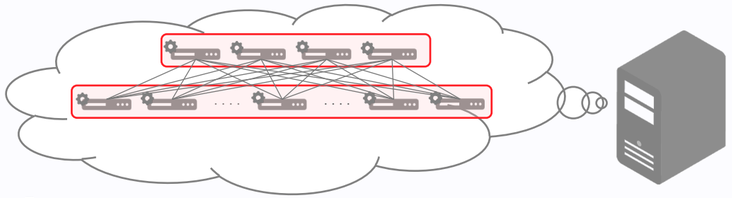
挑戦期

SiteOPS

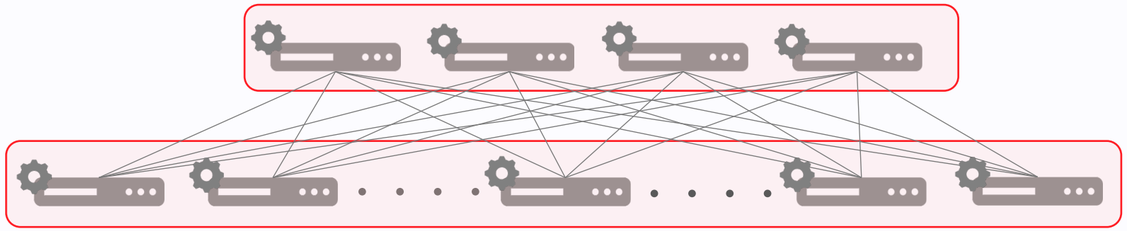
Yahoo! JAPAN Site Operations

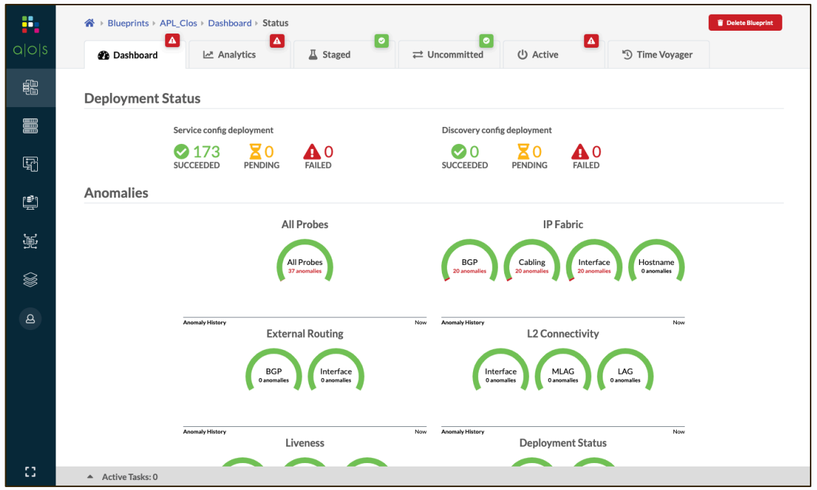
Gen3: Apstra AOS, 2017~

- 社内 Hadoop 基盤 第三弾
- 商用ツール Apstra AOS 導入
  - Intent Based な IP Clos 管理ツール
  - aos-agent 経由で制御
- NW機器
  - Spine: Arista Chassis, Leaf: Arista



AOS (Apstra OS)





AOS Web UI

Yahoo! JAPAN Site Operations

Gen3: Apstra AOSを導入して良かった点

ASN/IP 自動アサイン

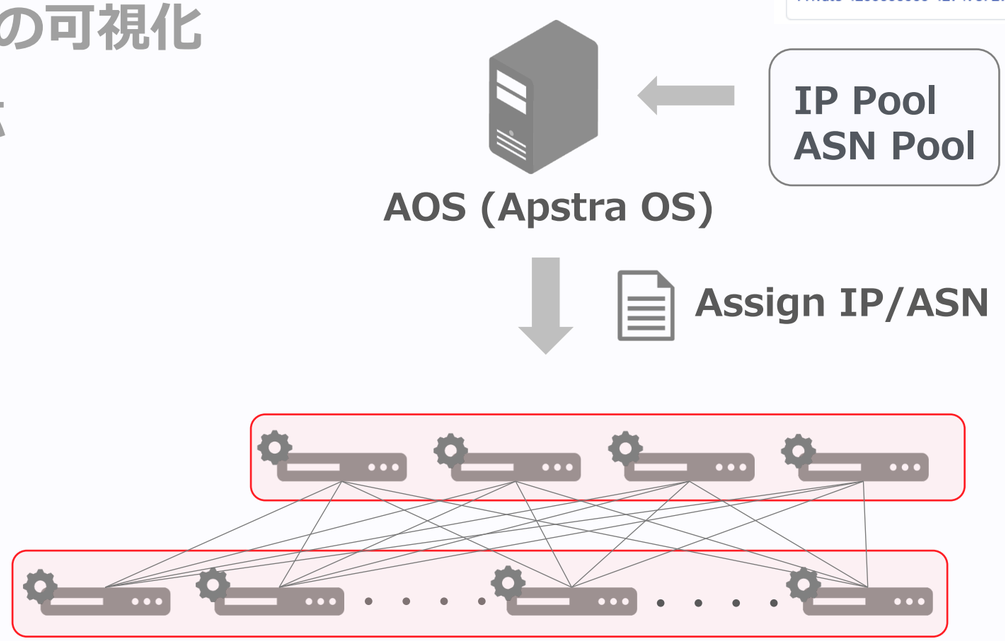
running-config の update をサポート

Config template を管理する仕組み (Configlet)

異常検知: 配線ミスの可視化

マルチベンダー対応

Pool Name	Ranges	Status
Leaf-AS	IN USE 4205308928 - 4205309439	IN USE
Spine-AS	IN USE 4205308440 - 4205308443	IN USE
Private-64512-65534	AVAILABLE 64512 - 65534	NOT IN USE
Private-4200000000-4294967294	AVAILABLE 4200000000 - 4294967294	NOT IN USE



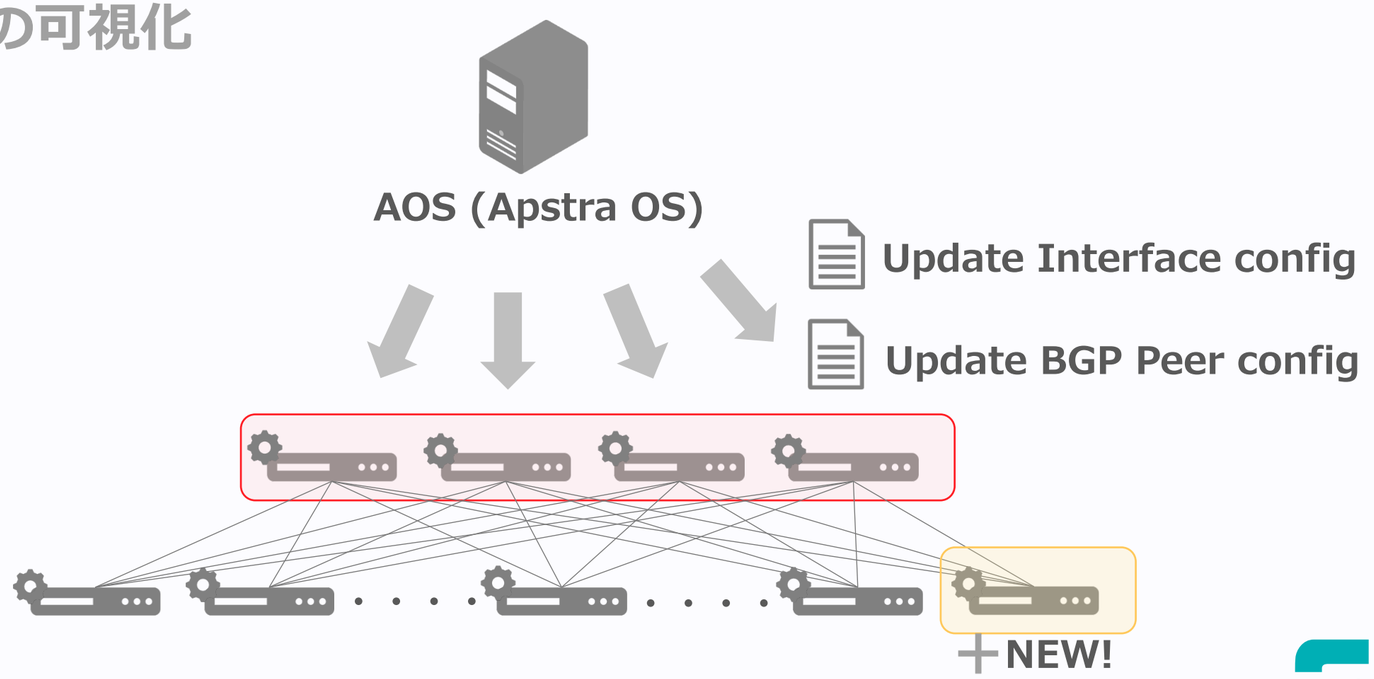
Gen3: Apstra AOSを導入して良かった点

- ✓ **ASN/IP 自動アサイン**
- ✓ **running-config の update をサポート**

Config template を管理する仕組み (Configlet)

異常検知: 配線ミスの可視化

マルチベンダー対応

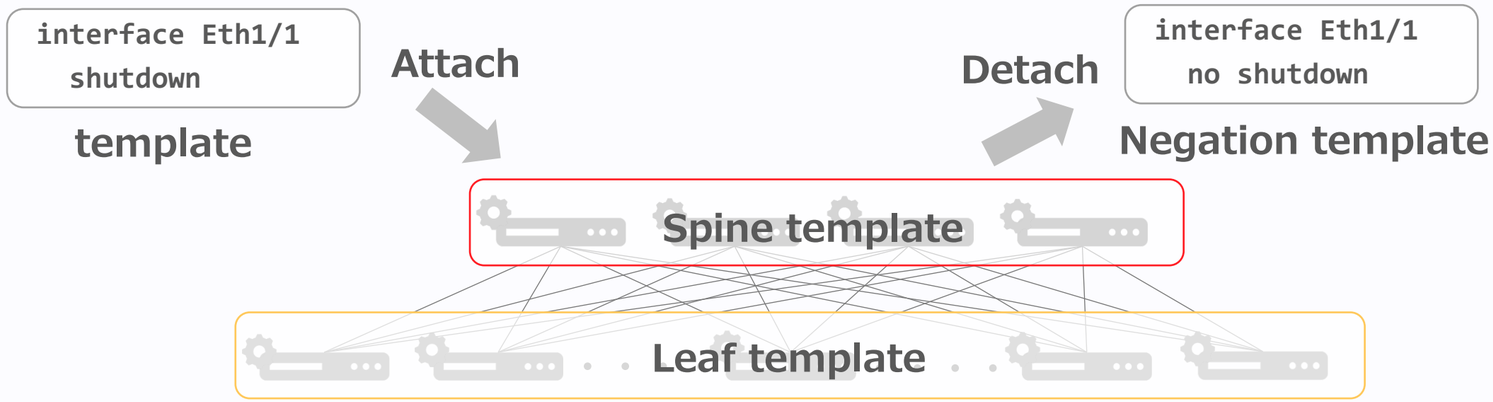


Gen3: Apstra AOSを導入して良かった点

- ✓ **ASN/IP 自動アサイン**
- ✓ **running-config の update をサポート**
- ✓ **Config template を管理する仕組み (Configlet)**

  異常検知: 配線ミスの可視化

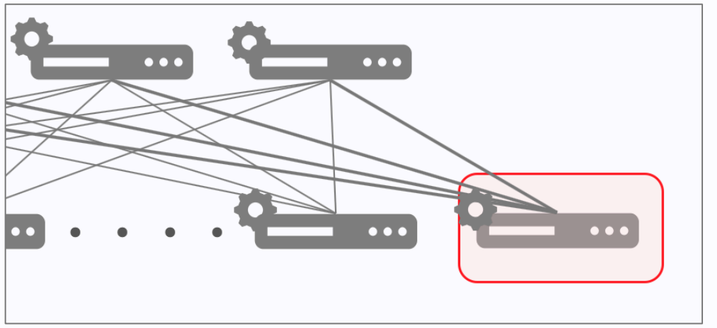
  マルチベンダー対応



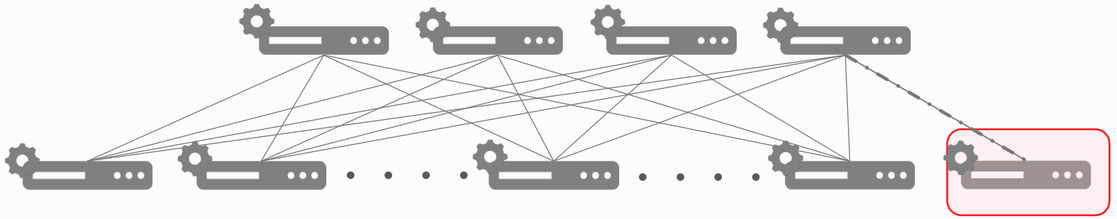
Gen3: Apstra AOSを導入して良かった点

- ✓ **ASN/IP 自動アサイン**
- ✓ **running-config の update をサポート**
- ✓ **Config template を管理する仕組み (Configlet)**
- ✓ **異常検知: 配線ミスの可視化**

マルチベンダー対応



Expected Connectivity



Actual Connectivity

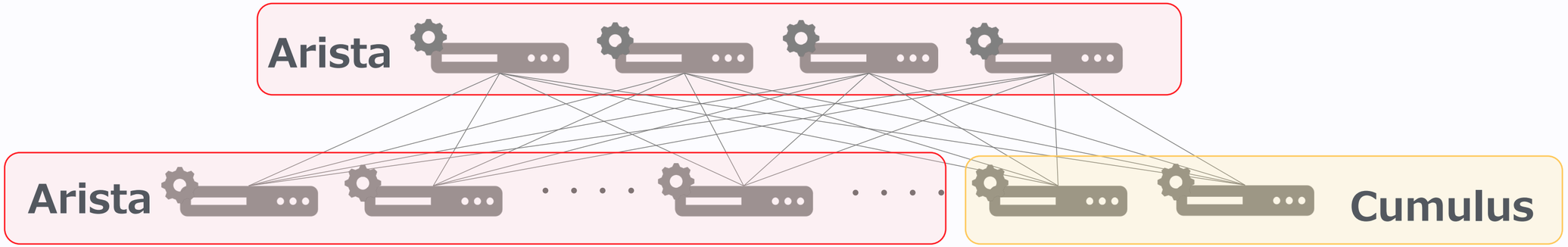
Assert

AOS (Apstra OS)

SiteOPS
Yahoo! JAPAN Site Operations

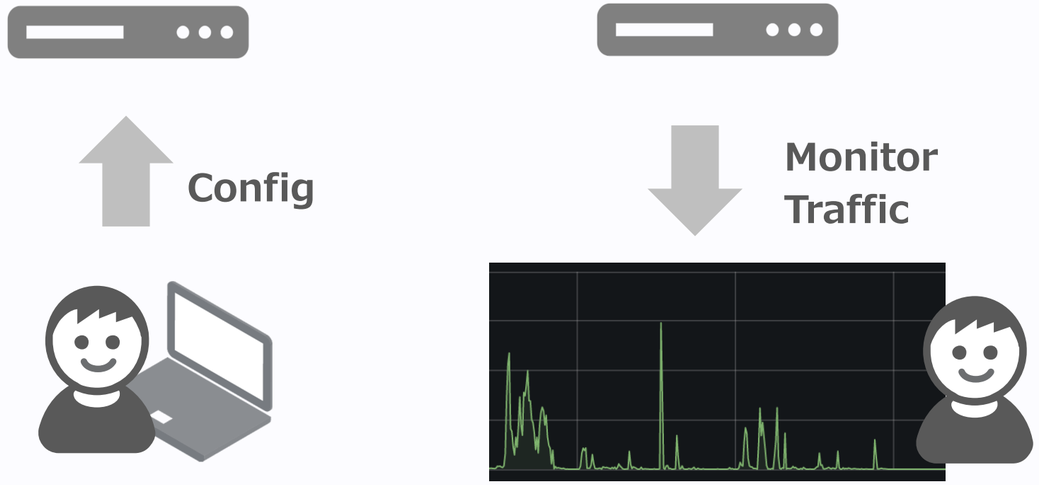
Gen3: Apstra AOSを導入して良かった点

- ✓ **ASN/IP 自動アサイン**
- ✓ **running-config の update をサポート**
- ✓ **Config template を管理する仕組み (Configlet)**
- ✓ **異常検知: 配線ミスの可視化**
- ✓ **マルチベンダー対応**

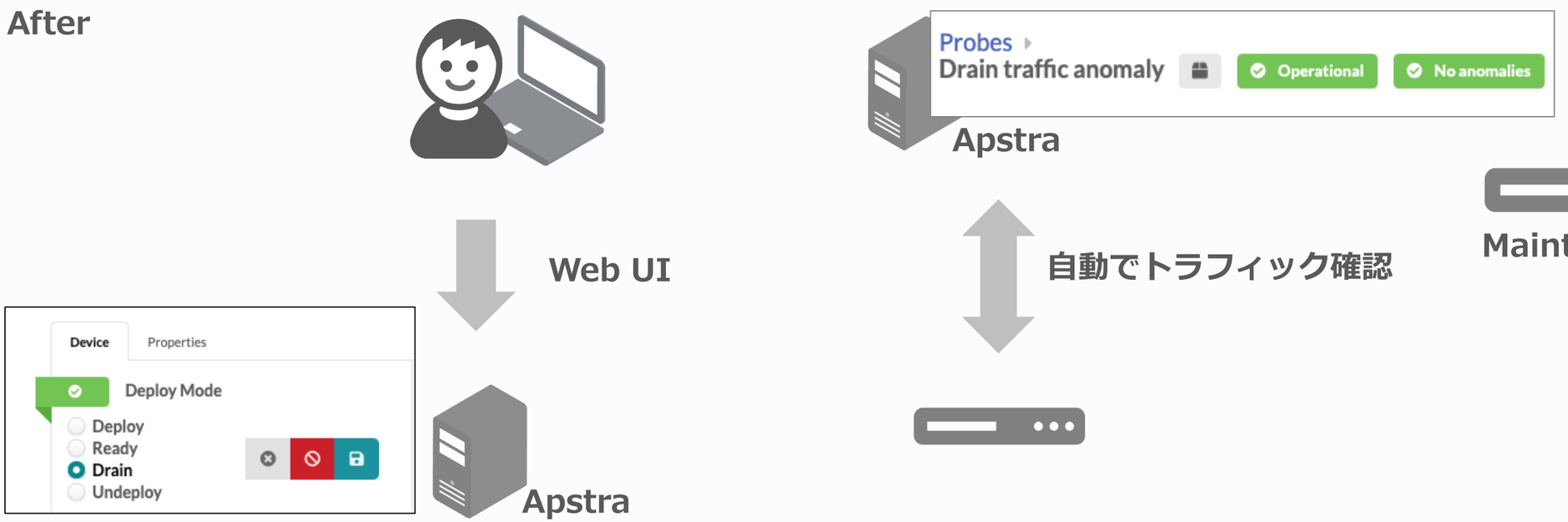


Yahoo! JAPAN Site Operations

トラフィック迂回作業が楽に



人が直接触る/見る手間を削減



SiteOPS
Yahoo! JAPAN Site Operations

Gen3: Apstra AOSでの課題

- 商用ツール
  - 世の中にドキュメントや知見が出回っていない
  - マニュアルはあるけど…

    操作法を覚えるのが大変 = 教育コスト

- 最新の NOS は使えない
- マルチベンダーに対応するため、NW機器の全機能が使えるわけではない
  - OSS を活用して新技術をすぐに取り入れられる選択肢も

ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

Gen1

2015

創世期

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen2

2016

拡大期

Apstra

- running-config更新
- マルチベンダ対応
- Intent-Based 異常検知
- 操作に慣れが必要
- OSS の選択肢も欲しい

Gen3

2017~

安定拡大期

Ansible

- 広く使われているOSS
- 自由なconfig
- Git で履歴/差分確認
- マルチベンダは難しい

Gen4.0/4.1

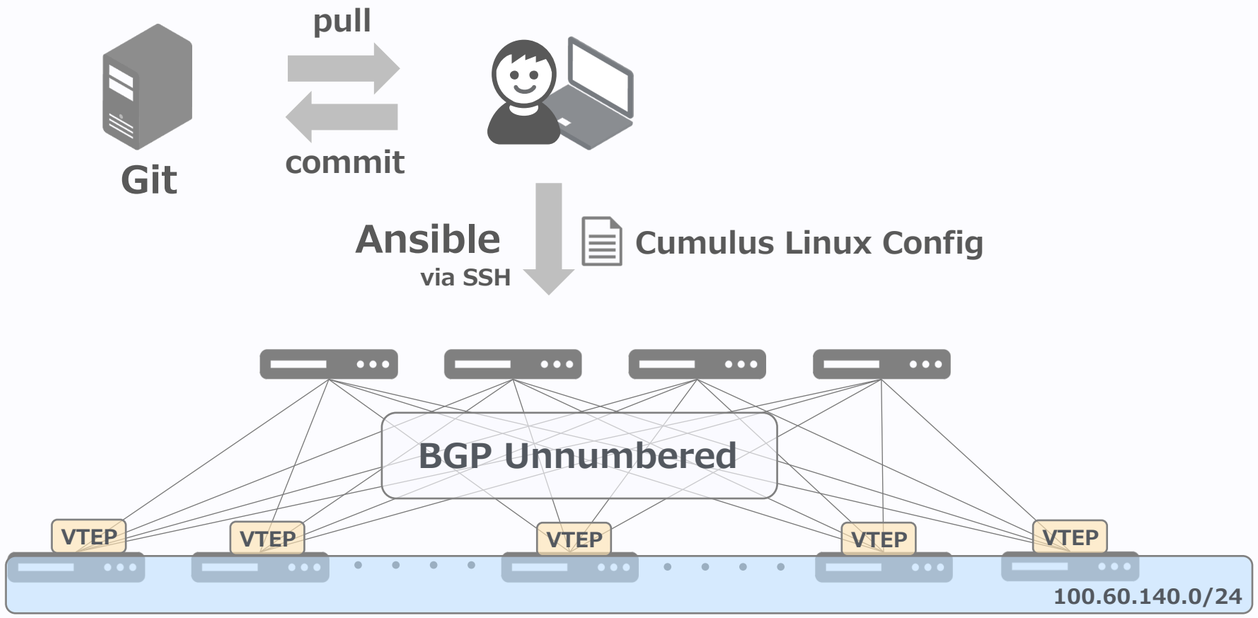
2017~

挑戦期

SiteOPS
Yahoo! JAPAN Site Operations

Gen4.0: Ansible, EVPN/VXLAN 2017

- 社内データ分析専用
- Ansible を導入
- NW機器
  - Spine: Whitebox Chassis (Cumulus)
  - Leaf: Whitebox (Cumulus)
- 新しい技術を取り入れ
  - BGP Unnumbered
  - EVPN/VXLAN で L2 延伸



Gen4.0: Ansible で良くなった点

✓ Ansible が OSS, 広く使われるツール

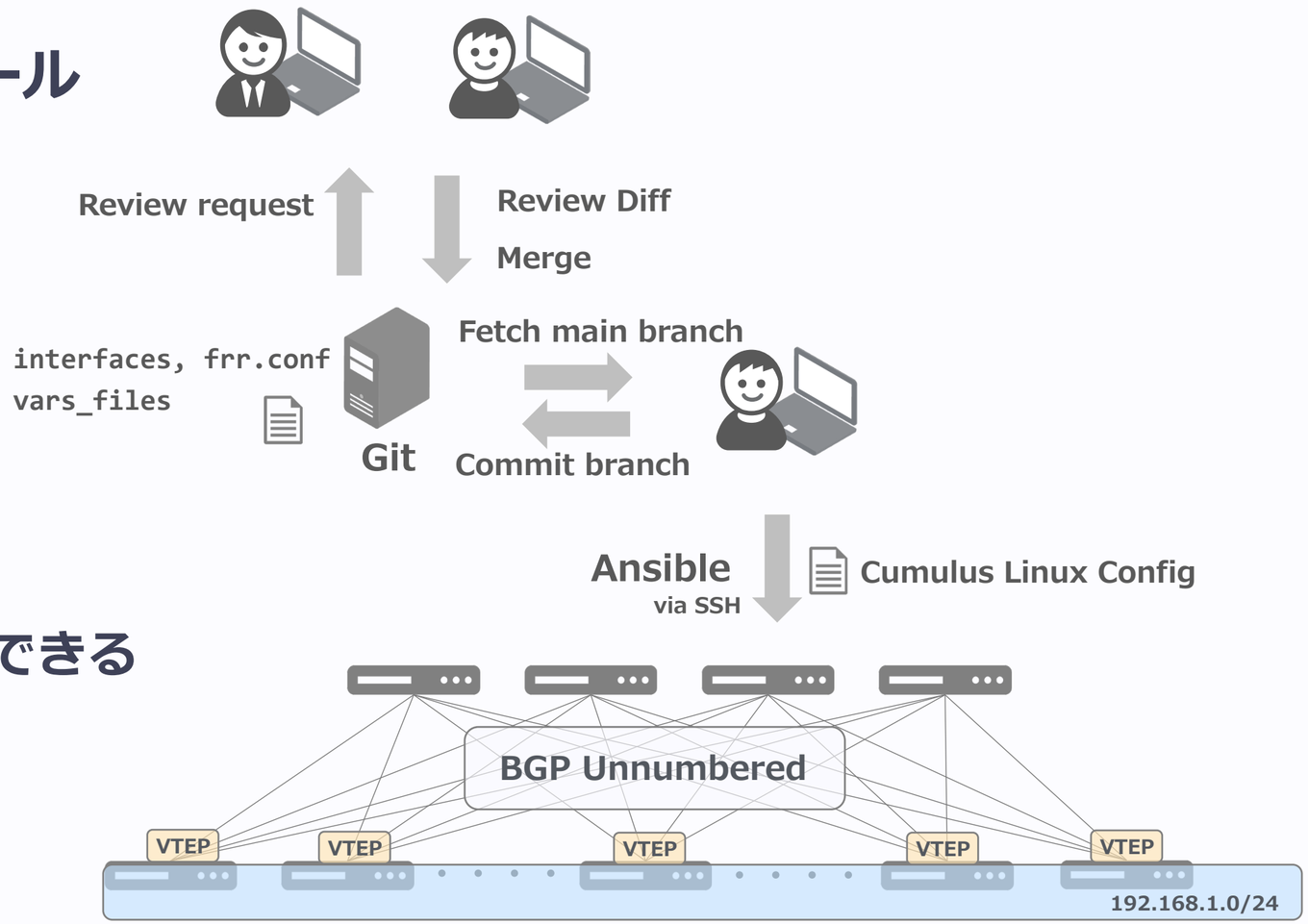
世の中に知見がある

簡単に使える

✓ Playbook の Git 管理で事故防止

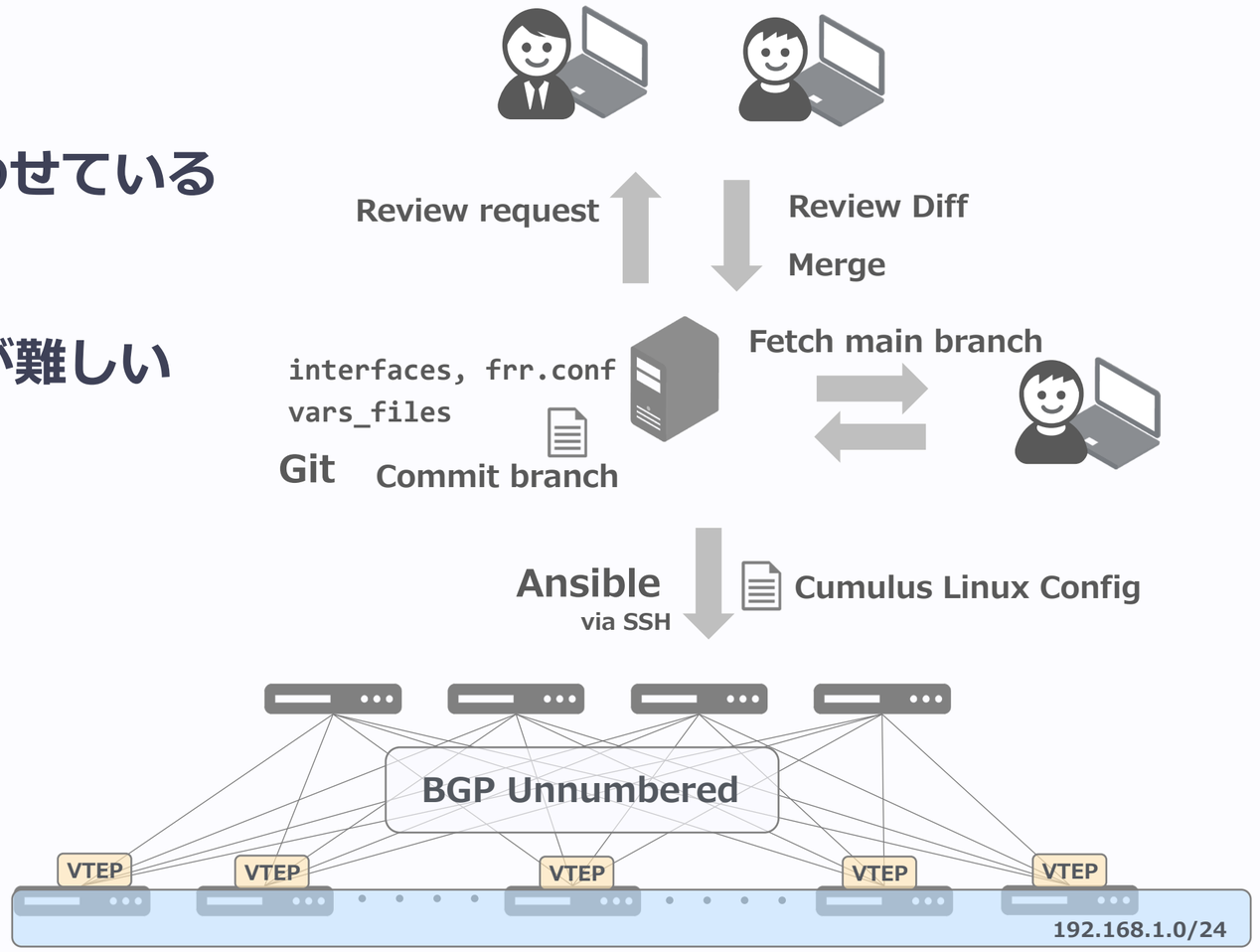
Deploy に PR でレビューする仕組み

危険な変更が無いか第三者がチェックできる



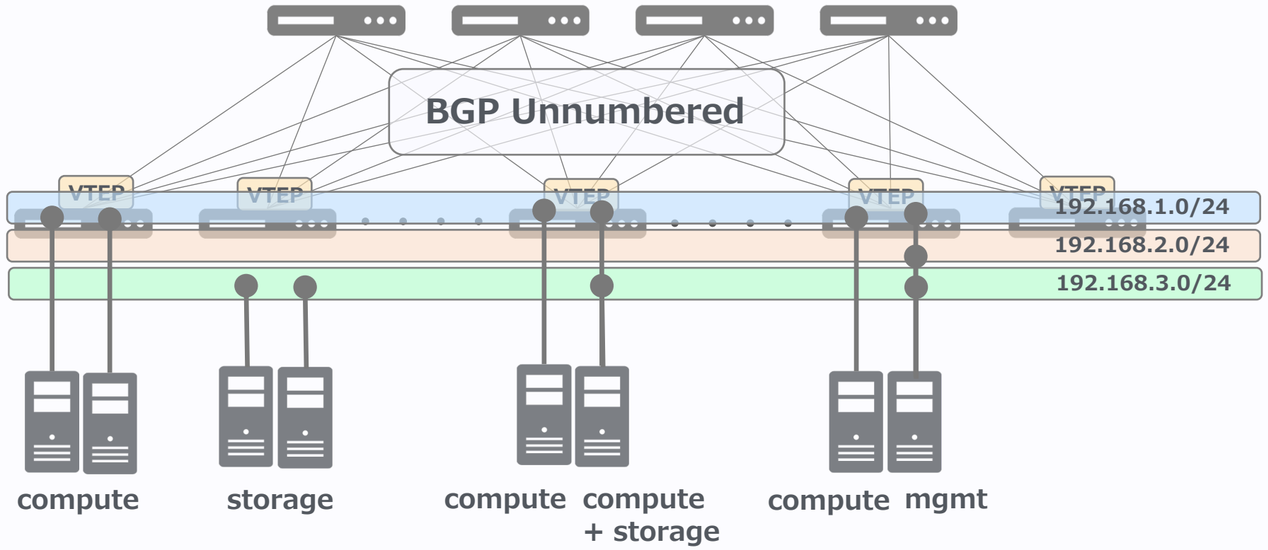
Gen4.0: Ansible での課題

- **Cumulus Linux での構築が前提**
  - **Linux の config template を組み合わせている (interface config/FRRouting)**
  - **他の NOS/Software との組み合わせが難しい**
- **異常検知の仕組みは無い**
  - **必要なものは自前で用意が必要**



Gen4.0: EVPN/VXLAN での課題

- **運用期間が長くなり, L2 構成が複雑に**
  - **Underlay network の品質や構成は同じ**
  - **Overlay network の管理が大変**
    - **Port毎に異なる access/trunk vlan の設定**

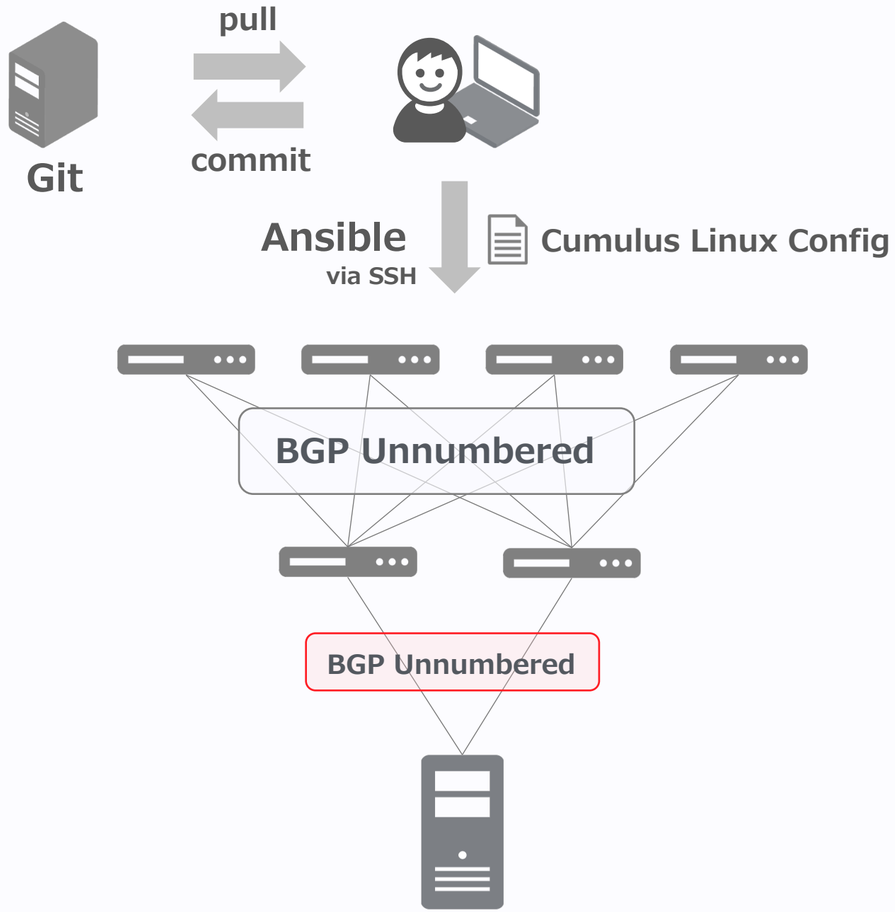



```
devices:
    lfc-1am01:
        profile: "LEAF"
        interfaces:
            lo:
                ipv4: "[redacted].109.26/32"
            eth0:
                ipv4: "[redacted].110.29/23"
                gateway: "[redacted].110.1"
            swp9:
                bridge-pvid: 1847
                bridge-vids: 1823
            swp10:
                bridge-pvid: 1820
                bridge-vids: 1823
            swp11:
                bridge-pvid: 1847
                bridge-vids: 1823
            swp12:
                bridge-pvid: 1820
                bridge-vids: 1823
            swp18:
                bridge-pvid: 1820
                bridge-vids: 1823
            swp19:
                bridge-pvid: 1847
                bridge-vids: 1823
            swp30:
                bridge-vids: 1823 1827 1846
            swp31:
                bridge-pvid: 1847
                bridge-vids: 1823 1827 1846
            swp32:
                bridge-vids: 1823
            swp37:
                bridge-access: 1821
            swp40:
                bridge-access: 1847
            swp41:
                bridge-access: 1847
            swp42:
                bridge-access: 1847
        bgp:
            bgpasn: "65001"
        bridge-pvid: 1820
        vrfs:
```

L2 管理コストをかけずにスケールアウトしたい

Gen4.1: Ansible, ALL-L3 IP Clos, 2019~

- **Private Cloud VM用**
- **Ansible**
- **NW機器**
  - **全て Box 型 Whitebox (Cumulus)**
    - **Chassis スイッチ → Boxスイッチ**
    - **大幅なコスト減**
- **EVPN/VXLAN 廃止**
- **サーバまで ALL-L3 接続**



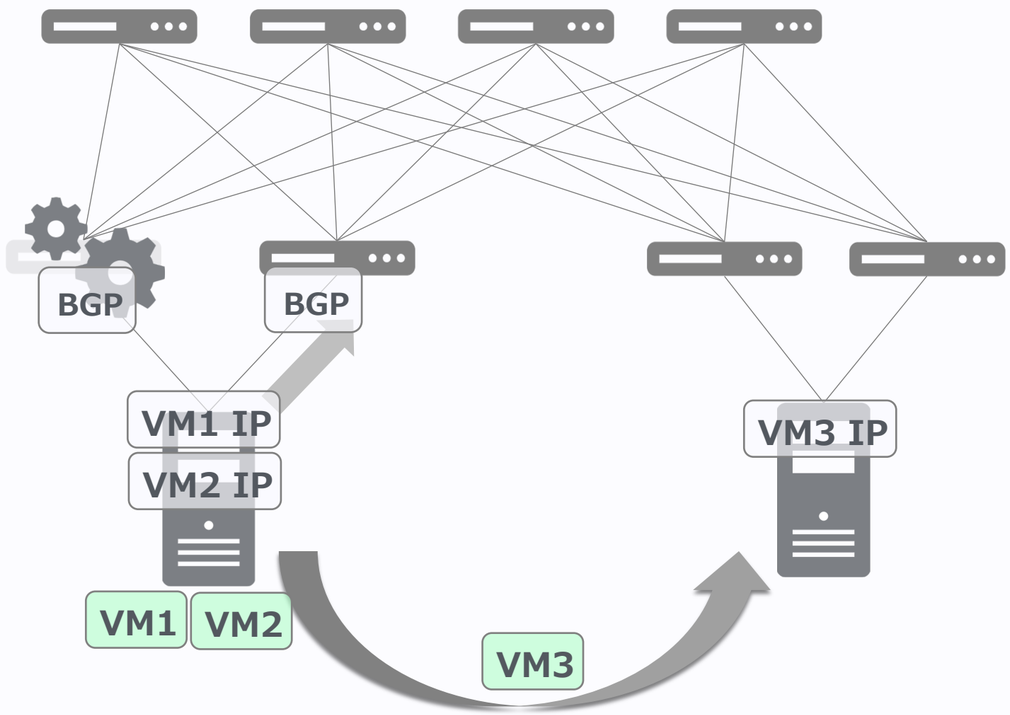
Gen4.1: ALL-L3 化で良くなった点

L3 接続でサーバの uplink 冗長が可能

- BGP だけでトラフィックを迂回できる
- L2冗長と比べると安定, 堅い
- メンテナンスの社内調整がしやすい

VM のラック間マイグレーション

- 冗長化された Leaf に /32 単位で VM の IP を広報
- VXLAN 不要

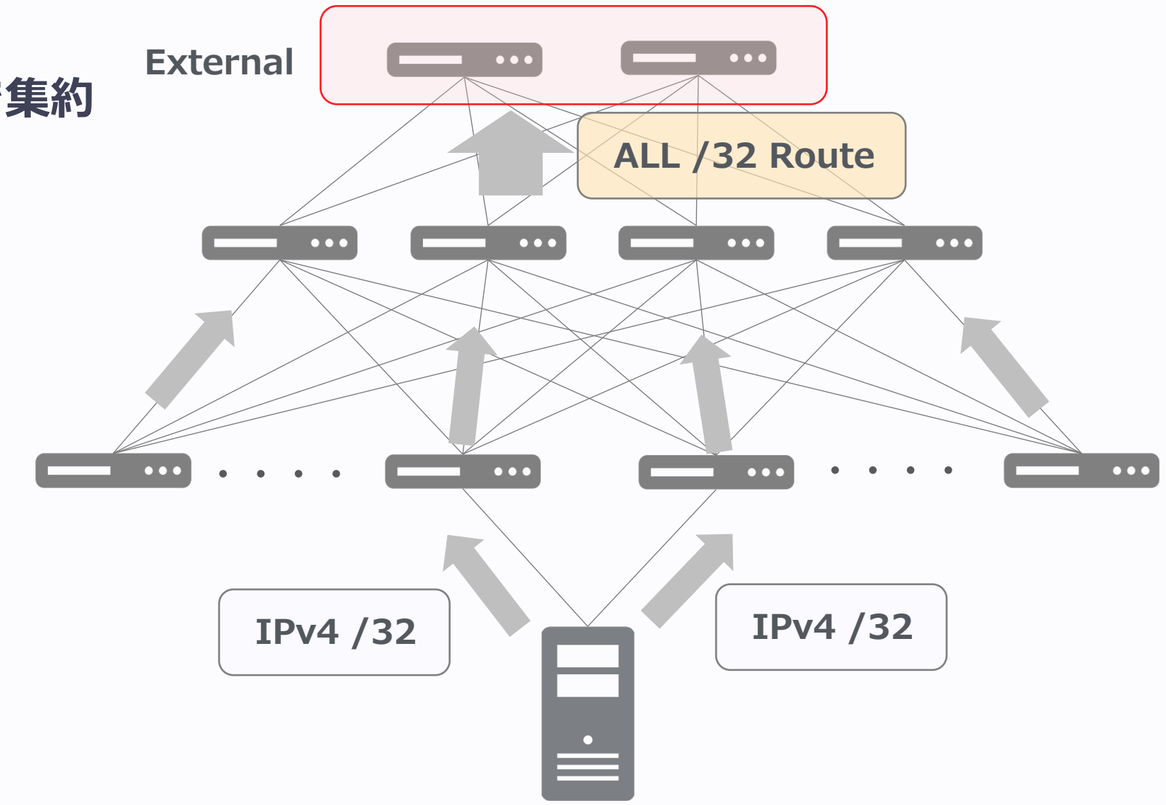


VM migration

Gen4.1: ALL-L3 化で困った点

- **Clos 網内の Route Table が肥大化**
  - **収容VM数 < External の Route Table サイズ**
  - **Clos 網内の IPv4 /32 route は External Router で集約**

- **収容数が増えて Route Table 逼迫してきたとき どうする？**
  - **External層ごと Clos Network を分離する**
  - **External層の Route Table をスケールアップ**



ヤフーの IP Clos 管理ツール

独自実装デプロイ

- 初期設定自動化
- 設定ミス防止
- 自社製デプロイスクリプト
- 手動リソースアサイン

Gen1

2015

創世期

OpenClos

- 内製からOSS活用へ
- 自動リソースアサイン
- Config 更新できない
- 他社製NW機器に非対応

Gen2

2016

拡大期

Apstra

- running-config更新
- マルチベンダ対応
- Intent-Based 異常検知
- 操作に慣れが必要
- OSS の選択肢も欲しい

Gen3

2017~

安定拡大期

Ansible

- 広く使われているOSS
- 自由なconfig
- Git で履歴/差分確認
- マルチベンダは難しい

Gen4.0/4.1

2017~

挑戦期

ツールの使い分け, Apstra V.S. Ansible

Apstra

- 複数の NOS を混合して使う
  - Arista EOS + Cumulus
- 自動でアドレスアサイン
- 異常検知の仕組み

- メンテナンスコストの削減
  - Drain mode を使って簡単に実現

Ansible

- Cumulus Only
- 新しい技術を取り入れ
  - ALL-L3

- メンテナンスコストの削減
  - g-shut community を ansible で投入

一方で… 少ない人数で運用するには

監視ツールは共通化されたほうが使いやすい

ネットワークの監視体制

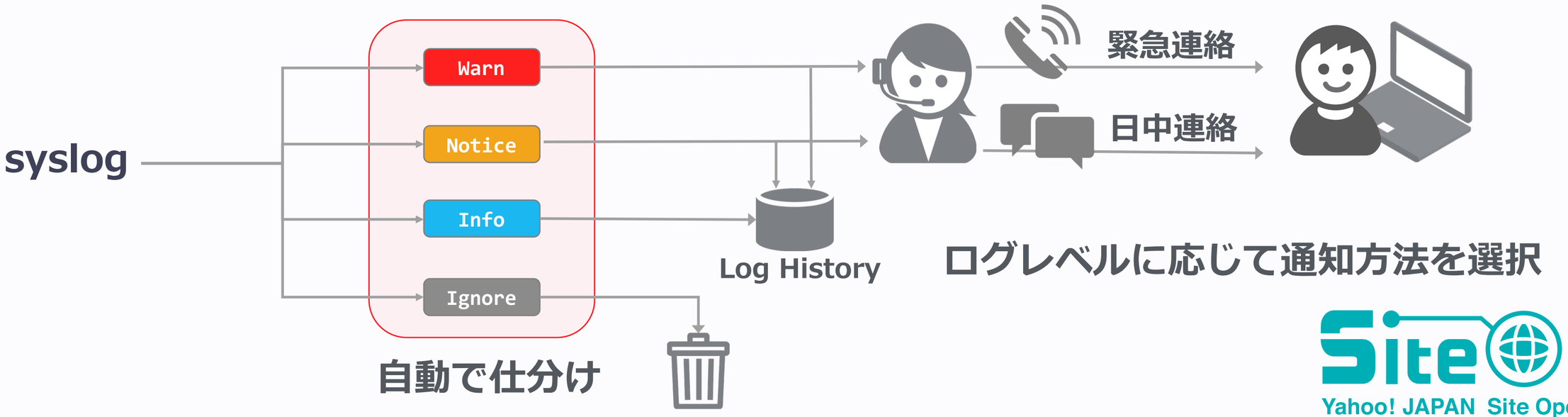
- 監視の仕組み
- Clos Network のアラート体制
- 日々の運用での課題

監視の仕組み - Syslog

Syslog ベースのイベント検知

事前に syslog をレベル別仕分け

Warn	watchfrr[18602]: [EC 268435457] bgpd state -> down : read returned EOF
Notice	bgpd[10014]: %ADJCHANGE: neighbor swp55(neighbor_hostname) in vrf default Down Interface down
Info	systemd[1]: Reloading FRRouting.
Ignore	switchd[2573]: sync.c:3802 IPv4 Route Summary (2553556) : 1 Added, 0 Deleted, 0 Updated, 0 Skipped in 23468 usecs



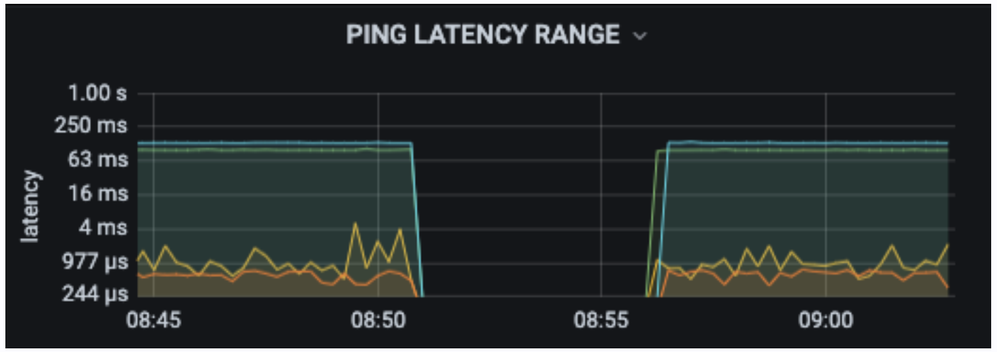
SiteOPS
Yahoo! JAPAN Site Operations

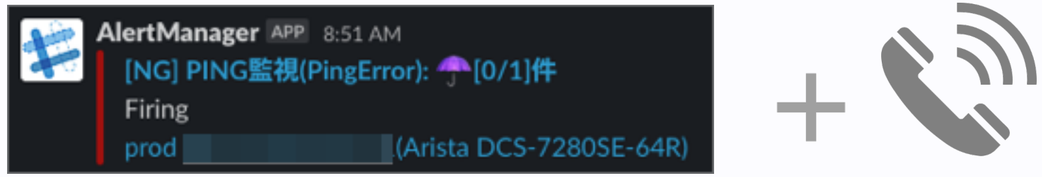
監視の仕組み - Prometheus

Prometheus を使った metrics 監視

詳しくは JANOG40
「フロントエンドエンジニアがY!のNW部隊で取り組んだこと」
https://www.janog.gr.jp/meeting/janog40/program/soft

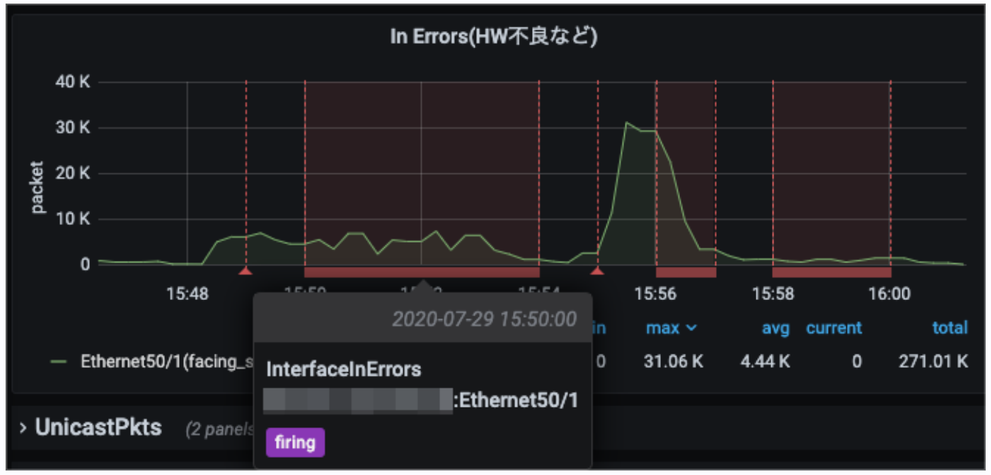
PING

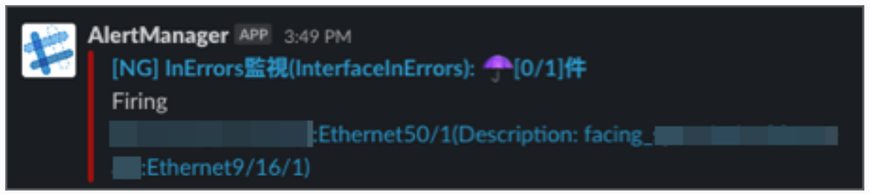




緊急度が高いアラートは通知+電話(夜間)

Interface Error





SiteOPS
Yahoo! JAPAN Site Operations

Clos Network のアラート体制

Warn 可及的速やかに回復が必要な場合: 深夜でも対応する

- **Ping Lost (= ダウンしたとき)**
  冗長化されていない Leaf スイッチが単一障害点になる場合
- **Interface Error**
  品質の低いリンクを shutdown して代替ルートへ迂回させたい

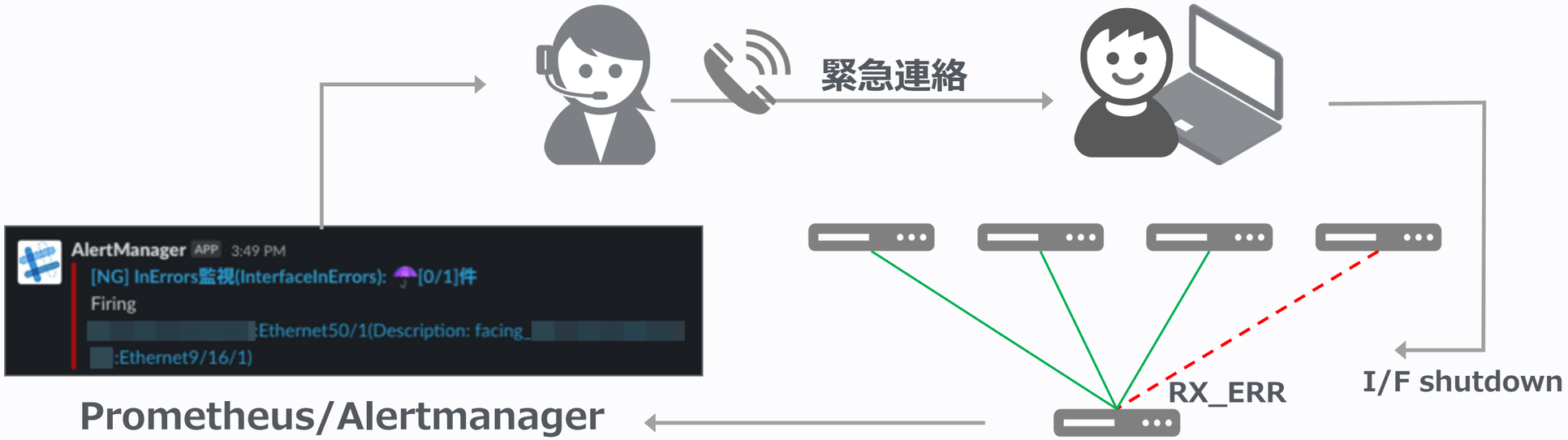
Notice 復旧を急がない場合: 翌日勤帯に対応する

- **BGP Peer down**
- **Interface down**

  Leaf uplink 4本中, 1-2本落ちても支障無い

Clos Network 日々の運用の課題

アラート発火後のオペレーションが手作業



大きな手間はかからないが, 確実に時間を取られる作業

今は運用を回せているが…

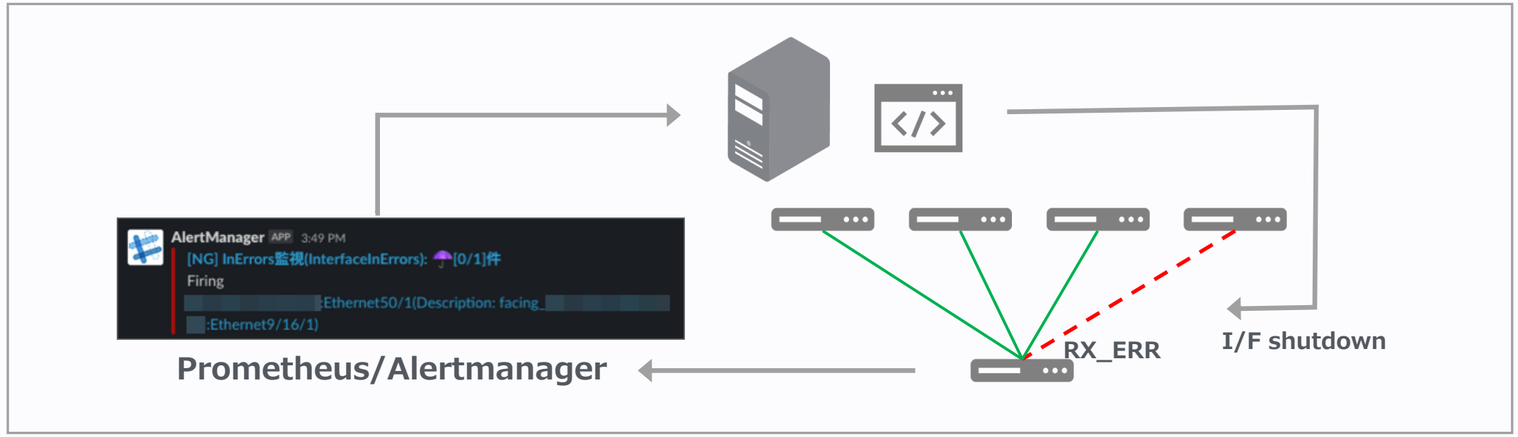
Clos Network 日々の運用の課題

ノード数が増えると故障も増える (特に配線回り)

- Clos Network ノード数: **1000+**
- 準備中ノード数: **500+**
- Clos Network 担当: **5人**

継続的な運用のために

障害時オペレーションの省力化 (=自動化) がこれからの課題



Yahoo! JAPAN Site Operations

まとめ

- IP Clos 管理ツール
  - 今は 2種類のツールを並行して使っている
    - Apstra – マルチベンダー
    - Ansible – Cumulus Linux
  - 要件毎に使いわけ
- 監視
  - 2つの観点で監視
    - Syslog – Event 検知
    - Prometheus – Metrics
  - なるべく日中にアラートをもらうように
  - 障害対応の自動化が課題

Discussion

どのように構成を管理していますか？

運用していて困ったことなど