



JUNOS Hands On Training “EX / QFX” Course

Juniper Network, K.K.

2022 年 10 月 rev. 2.1

はじめに

- 本資料にあるロードマップの内容は、資料作成時点におけるジュニパーネットワークスの予定を示したものであり、事前の通告無しに内容が変更されることがあります。
- またロードマップに描かれている機能や構成は、購入時の条件になりませんので、ご注意ください。

Legal Disclaimer:

This statement of product direction (formerly called “roadmap”) sets forth Juniper Networks' current intention and is subject to change at any time without notice. No purchases are contingent upon Juniper Networks delivering any feature or functionality depicted on this statement.



Junos Basic

トレーニング概要 「Junos Basic」

トレーニング内容（前半）	記載ページ
ジュニパーネットワークス会社紹介	P. 6
Junos とは	P. 12
運用面からみた Junos のアドバンテージ	P. 22
トレーニング・デバイスへのアクセス方法	P. 27
CLI モードと各モード間の移動	P. 31
Junos CLI 操作 ～ Operational モード～	P. 38
Junos CLI 操作 ～ Configuration モード～	P. 57
Junos システム設定	P. 74
Junos インタフェース設定	P. 82
Junos 経路設定	P. 90
Firewall Filter の設定	P. 94

トレーニング概要 「Junos スイッチ “EX / QFX” コース」

トレーニング内容（後半）	記載ページ
Junos EX シリーズ製品紹介	P. 103
LAB.1 Junos の基本的な操作・設定	P. 112
LAB.2 Interface の設定	P. 127
LAB.3 Routing の設定	P. 140
LAB.4 Firewall Filter の設定	P. 147
Virtual Chassis とは	P. 153
Virtual Chassis Deep Dive	P. 165
LAB.5 Virtual Chassis の設定	P. 184
Appendix	P. 202



ジュニパーネットワークス 会社紹介

ジュニパーネットワークス 会社概要

設立：1996年2月（1999年3月）

本社所在地：カリフォルニア州サニーベール

Juniper Networks（NYSE: JNPR）

（ジュニパーネットワークス株式会社）

CEO：Rami Rahim

（日本法人 代表取締役社長：古屋 知弘）

事業概要：ネットワーク機器（ルータ、スイッチ、ファイアウォール、
無線AP等）の製造・販売



ジュニパーネットワークスの戦略

Vision : ネットワークイノベーションにおけるリーダー

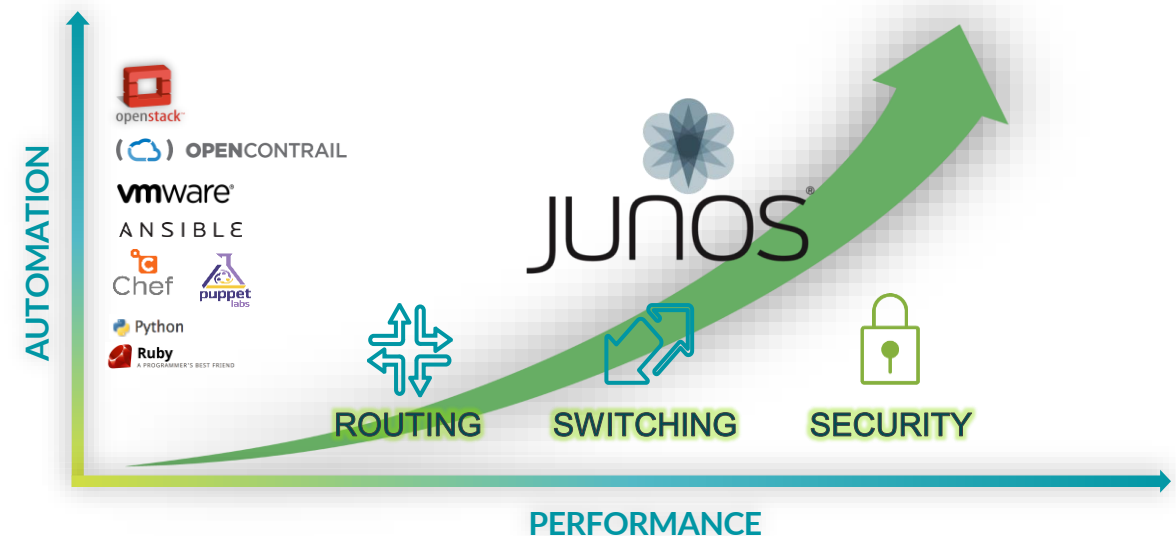
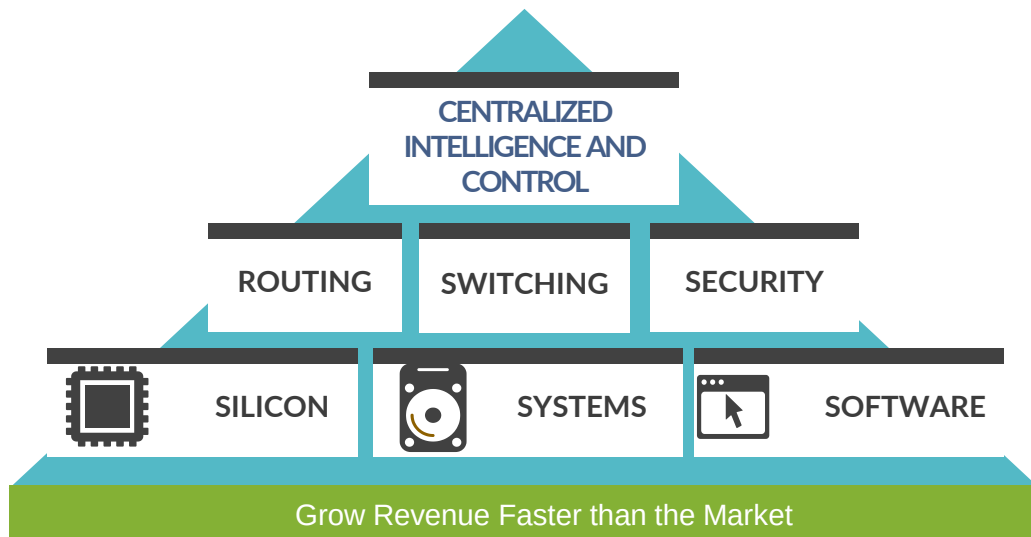
Go-To-Market : ハイパフォーマンスネットワーキングをビジネスの基盤と位置付けるお客様とパートナー様に価値を提供

パフォーマンスと自動化におけるバリュー

- ✓ スケーラビリティ
- ✓ 高コスト効率

- ✓ 信頼性
- ✓ セキュリティ

- ✓ 俊敏性
- ✓ 高効率



プロダクト・ポートフォリオ（カテゴリ別）

ROUTING



MX Series

MX10008
MX2020
MX2010
MX2008
MX960
MX480
MX240
MX150
MX104
vMX



PTX Series

PTX5000
PTX3000
PTX1000



ACX Series

ACX5000
ACX4000
ACX2100
ACX2000
ACX1100
ACX1000
ACX500

SWITCHING



EX Series

EX9250
EX9200
EX4650
EX4600
EX4400
EX4300
EX4100
EX3400
EX2300



QFX Series

QFX10016
QFX10008
QFX10002
QFX5700
QFX5220
QFX5210
QFX5200
QFX5130
QFX5120
QFX5110
QFX5100

SECURITY



SRX Series

SRX5800
SRX5600
SRX5400
SRX4600
SRX4200
SRX4100
SRX1500
SRX380
SRX345
SRX340
SRX320
SRX300
vSRX



NetScreen Series

NetScreen-5200
NetScreen-5400

JUNOS

SSG Series

SSG550M
SSG520M
SSG350M
SSG320M
SSG140



ISG Series

ISG2000
ISG1000



Network Director
Security Director

JUNOS: THE POWER OF ONE INTEGRATED ARCHITECTURE

Datacenter Service Gateway
SRX series



Universal Edge Router
MX series



Datacenter Fabric Switch
QFX series



JUNOS: THE POWER OF ONE INTEGRATED ARCHITECTURE

Branch Service Gateway **SRX series**



Campus Ethernet Switch **EX series**





Junos とは

「複数 OS」 対 「“One” のアプローチ」



ASA
IOS
IPS
OS-XE
NX-OS
IOS-XR

プラットフォーム毎に異なる OS と機能セット



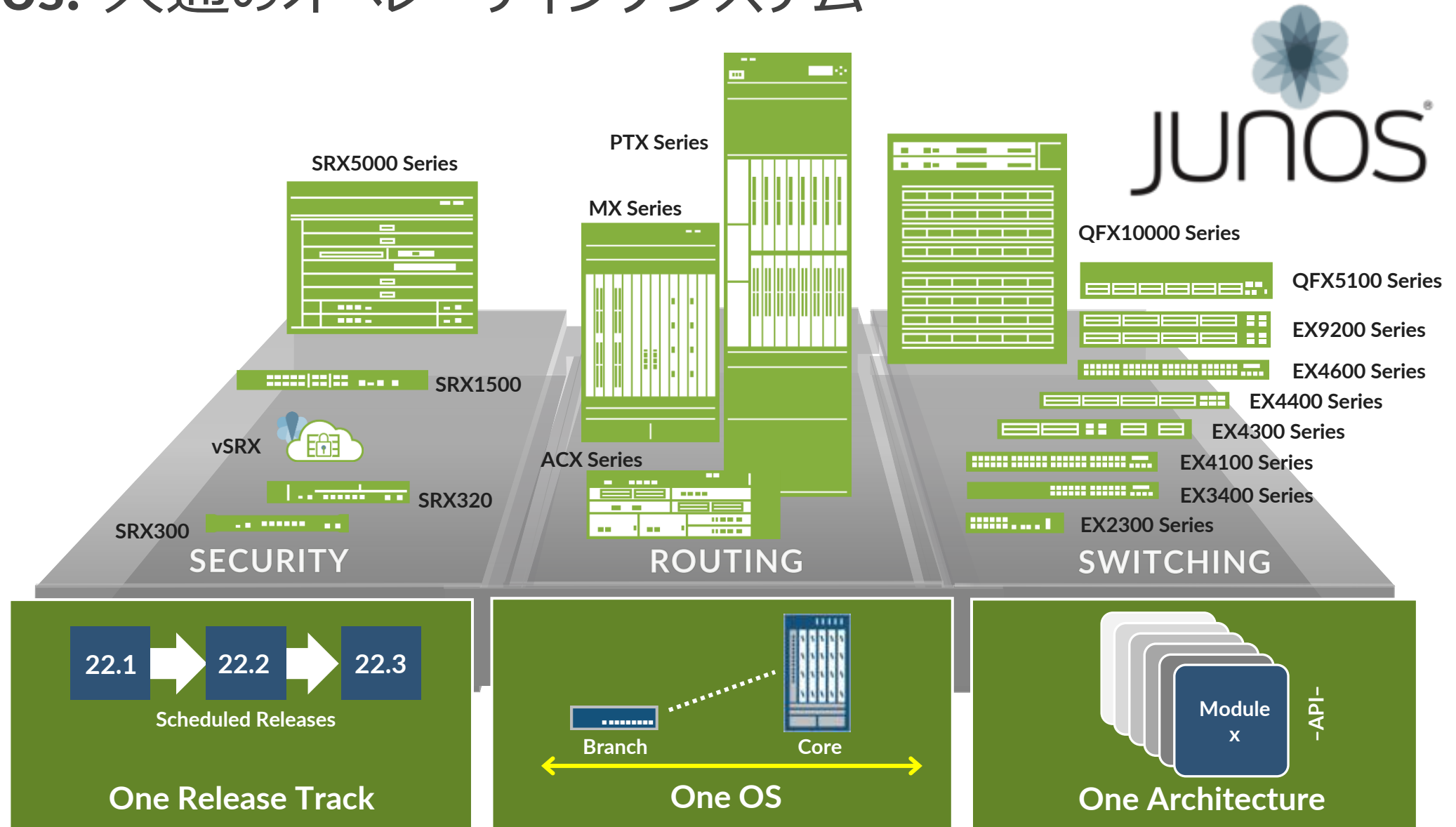
JUNIPER
NETWORKS

JUNOS

セキュリティもネットワークもカバーする
業界唯一のシングル・ネットワーク OS



Junos: 共通のオペレーティングシステム



「One」の強み

LEARN ONCE、INTEGRATE ONCE、QUALIFY ONCE

プラットフォーム共通機能

- Routing
- Layer 2 Switching
- Class of Service
- IPv4 and IPv6
- Etc...

Cross-Portfolio Commonality

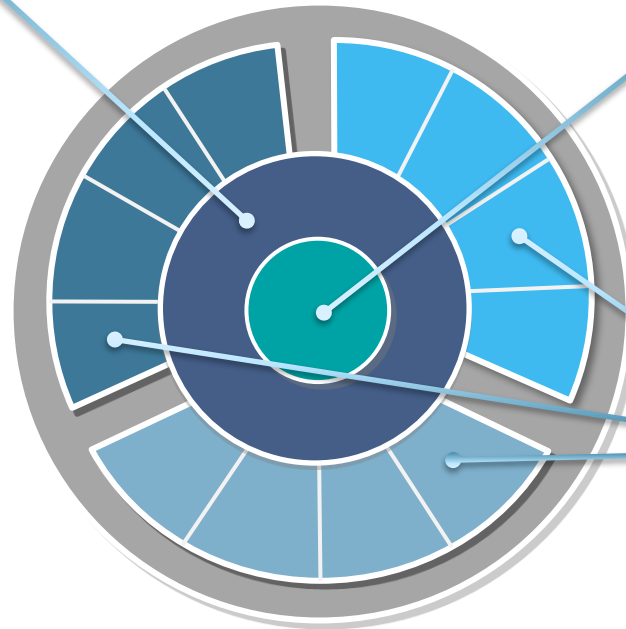
BGP/MPLS Control Plane

End-to-end Security

In-network Automation

SDK and Licensing of Junos

etc...



ベース・コンポーネント

- Kernel and μ Kernel
- Chassis Management (chassisd)
- IP Services (Telnet, SSH, NTP)
- Network Management
 - (AAA, CLI/mgd, XML/DMI, syslogd)

プラットフォーム専用機能

- Advanced Security (SRX)
- Virtual Chassis (EX/QFX)
- MPLS/EVPN (MX)
- ISSU (MX&EX9k)
- Etc...

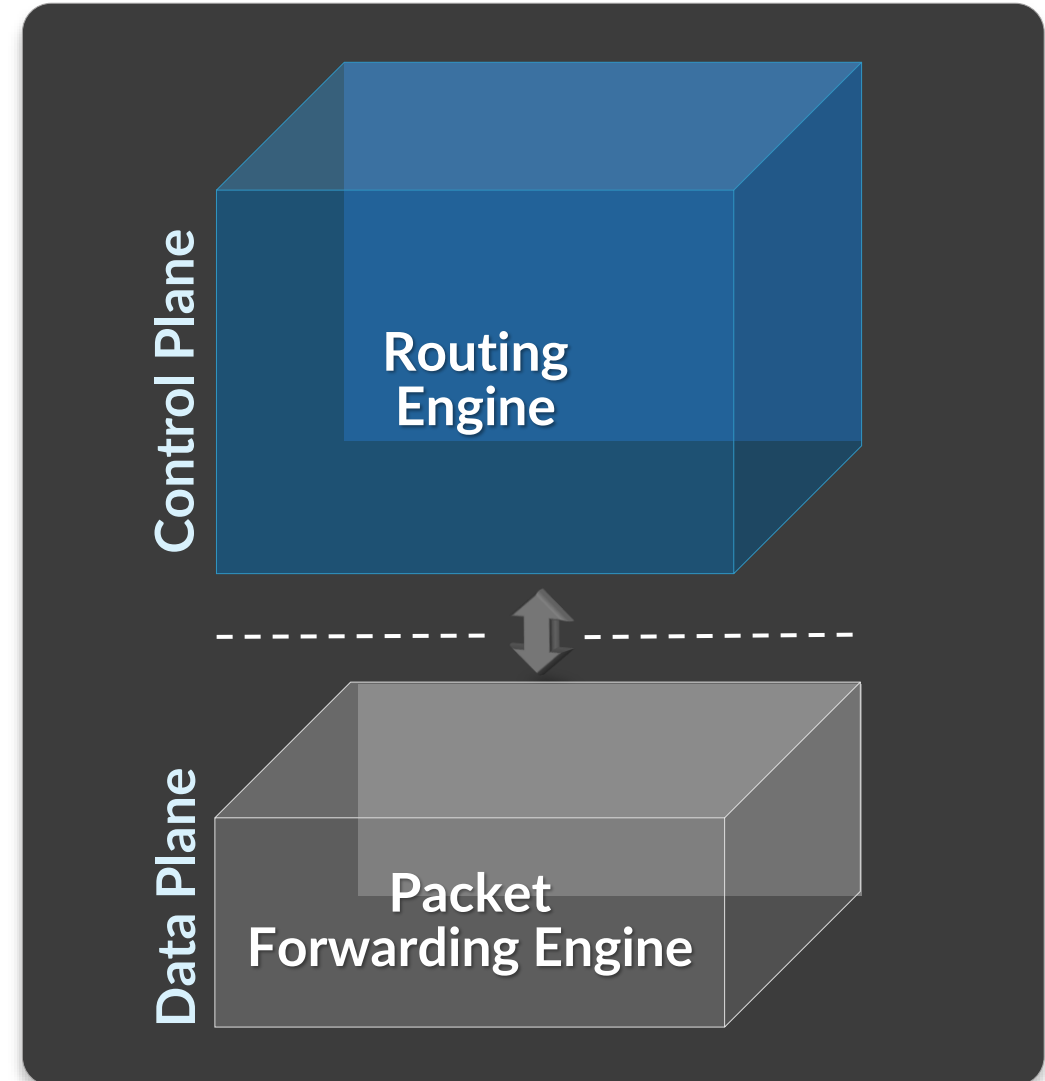
コントロールプレーンとフォワーディングプレーンの分離

Scale and Performance

- 各 **Plane** におけるパフォーマンスを担保
- より高いパフォーマンスをそれぞれの領域で独立して開発することが可能に

Resilient (※弾力性/復元力)

- 独立したオペレーション
 - **Routing Engine (RE)**
 - **Packet Forwarding Engine (PFE)**
- 冗長化に対するさまざまなオプションをそれぞれに提供



Junos の「 One 」アーキテクチャの進化

モジュラー型

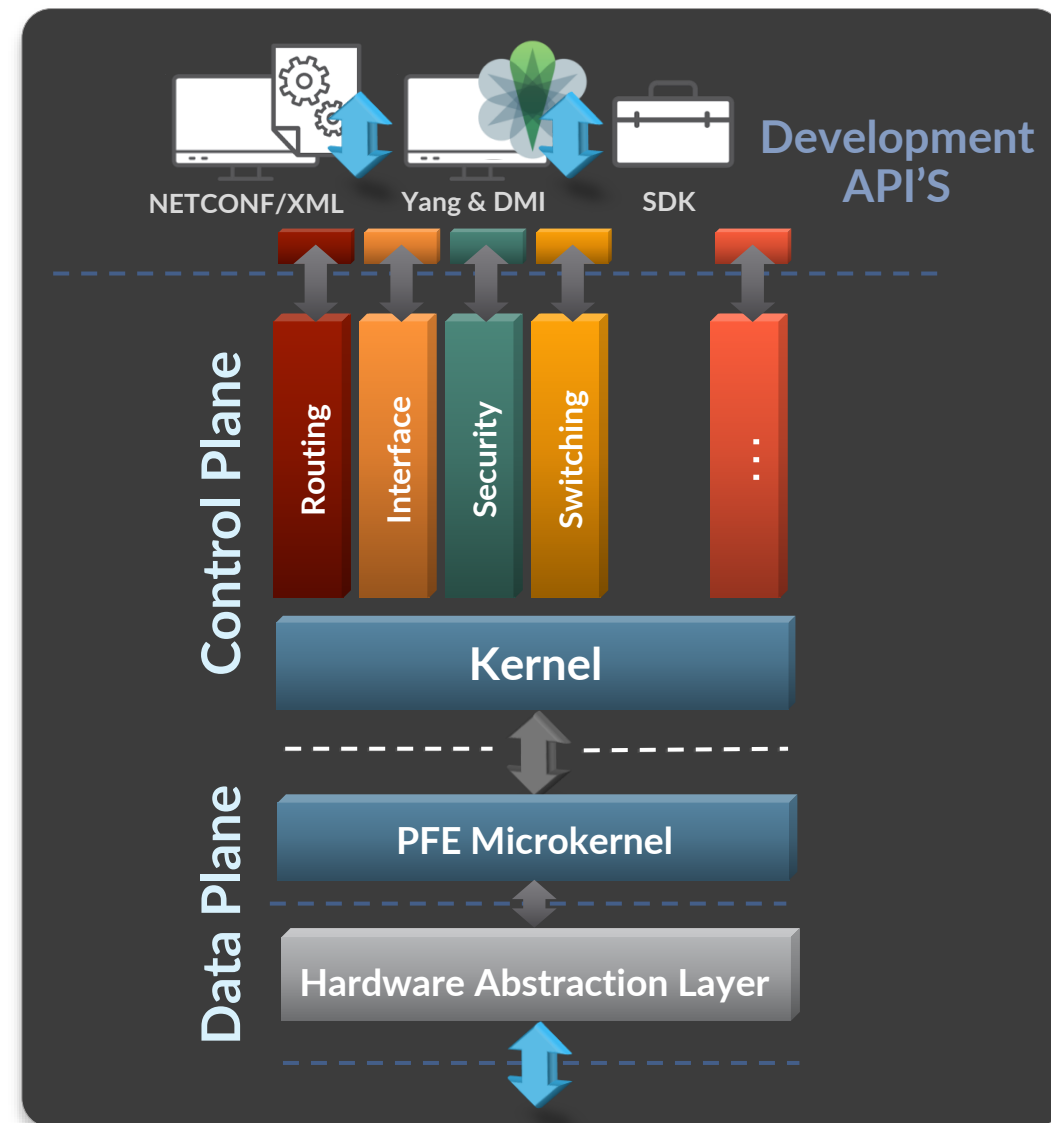
- 拡張性とパフォーマンスを担保するコンポーネント
- 冗長性、安定性、サービス拡張を効率的に提供するための独立したオペレーション

Scalable (※拡張性)

- Up: マルチコア & 64-bit
- Down: モジュールごとのパッケージング

Open (※オープン性)

- FreeBSD ベース
- API 連携、Junos 開発ツール (SDK)



Junos のアプローチ・運用者/設計者にとって

ネットワーク停止の原因に対する調査

計画停止

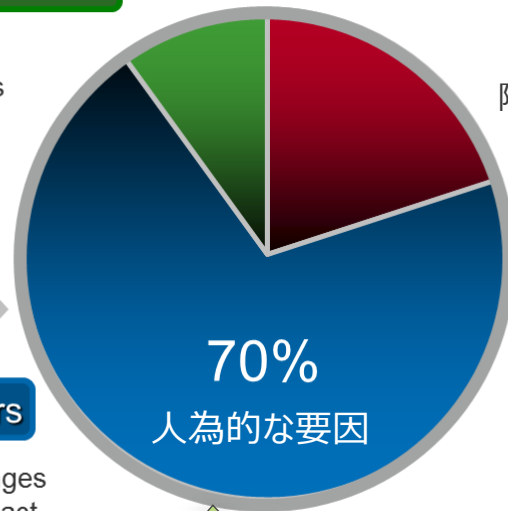
Planned Maintenance

Hardware and software upgrades

Junos Automation

Human Factors

Configuration changes that negatively impact network performance



70%
人為的な要因

障害やバグによる予期せぬ停止

Unplanned Events

Network failures, hardware events and software defects

全体の 70% は人為的なミスが原因でネットワークに悪影響

Junos の CLI は、業界標準型 CLI と根本的に異なるアプローチを採用

業界標準型 CLI

コマンドは 1 行毎に実行され、変更は即時反映される

ミスや間違いも即時反映
致命的な影響となることも...

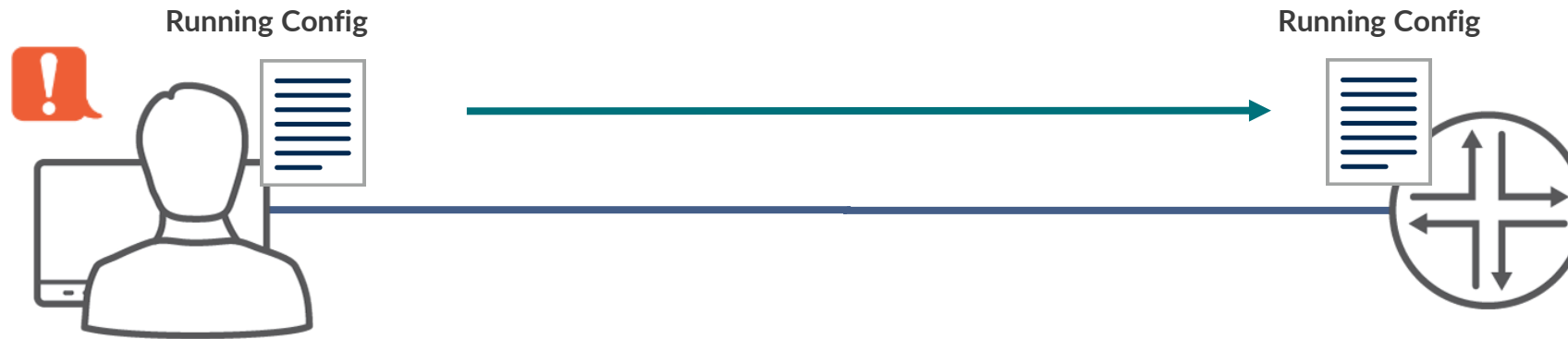
Junos CLI

コマンドは編集用ファイルのみ変更し、変更は意図したタイミングで反映させる

ミスや間違いがあっても
確認・修正してから適用

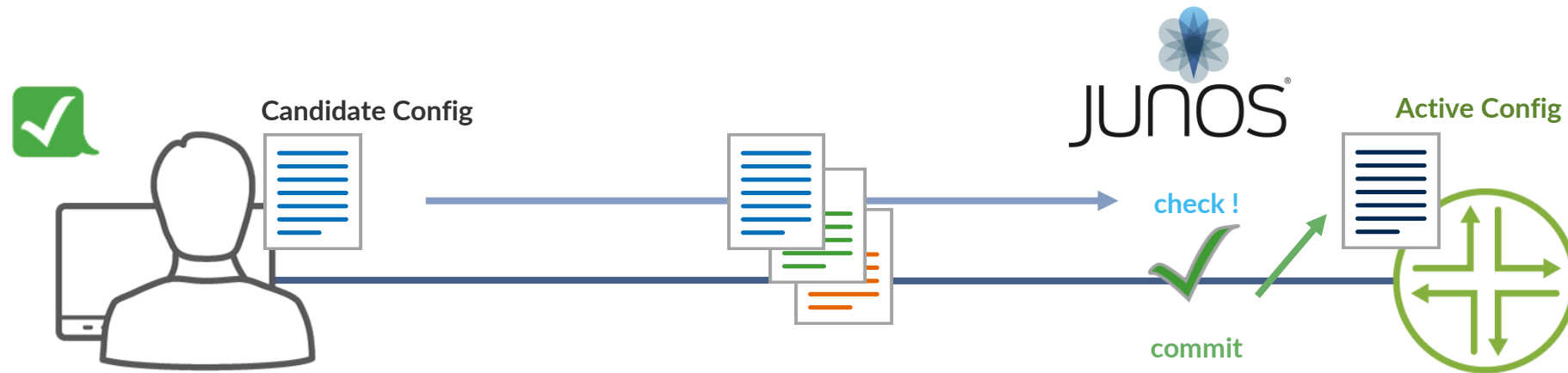
作業者の努力にたよるのではなく
ソフトウェアの仕組みでミスをなくすサポート

これまでの一般的な NW-OS の不便さ



- 一般的なネットワーク OS の場合、管理者がコンソールなどで設定変更を行う際、投入した設定が**即座に実稼働のネットワーク設定へと反映**されてしまう
- このことにより、
 - **ヒューマンエラーが発生する余地がある**
 - **設定の復旧が困難**
 - **意図しない設定を行ってしまうと、機器への通信自体が不可能になってしまうケースがある**などの課題が存在する

Junos の場合



- **Junos** の場合、管理者が設定変更を行うのは、あくまで **設定ファイル**
これを実ネットワークの設定へと投入するためには **Junos** によるシステムチェックを行った後に、
“**commit**” というコマンドを投入することにより反映させる
- この仕組みにより、
 - **Junos** のシステムチェックによる **ヒューマンエラーの予防**
 - 設定ファイルは過去 50 世代まで自動保存されるため、**一瞬で過去の状態へと戻すことができる**
 - 作成した設定ファイルを、“**ためしに**”投入してみることも可能
などのメリットを享受することができる

Junos のアプローチ : Human Factors への対応

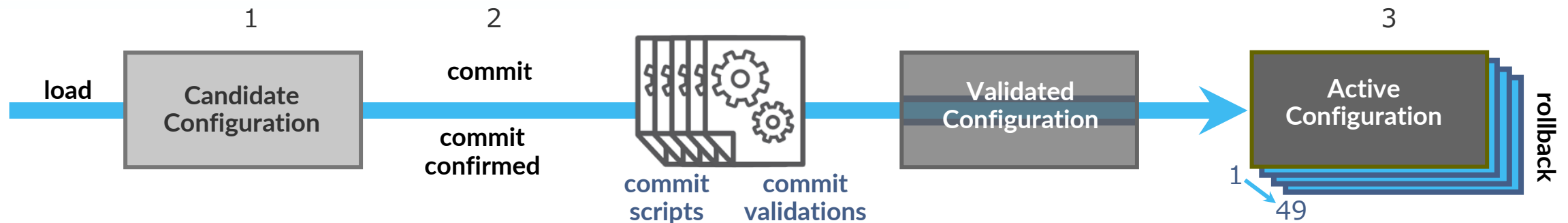
有効な Junos ツール

- “commit”
 - 設定変更を有効にするコマンド
 - 有効時に **Config** チェックをおこない、誤り（矛盾）がなければ投入した設定が有効となる
- “rollback”
 - 設定の履歴管理、設定・OS の切り戻しを容易に
 - 既存 **Config** を含み最大 50 世代までの管理が可能
 - “load” コマンドにより外部から設定ファイルを更新することも可能
- “JUNOScript” & “Event Policy”
 - スクリプティングによる自動化ツール
 - イベントをトリガーとした自動化機能

Benefits

Config ミスによるダウンタイムの回避

Config 変更/ 切り戻し作業の時間短縮





運用面からみた Junos の アドバンテージ

導入、運用、トラブルシュー트에有効な Junos Utility 群

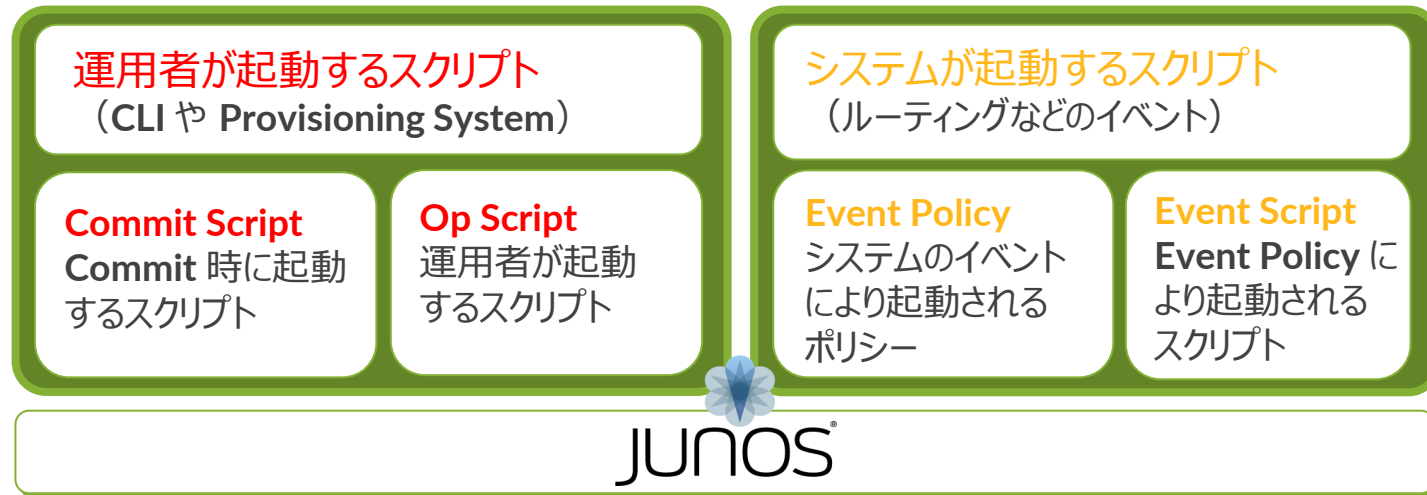
Junos は導入、運用、トラブルシュー트에有効な様々なツールを提供

- **Commit**
 - 設定変更を有効にするコマンド
 - **check**、**confirmed**、**compare** など様々な **Option** が使用可能
- **Rollback**
 - 設定の履歴管理、切り戻しを容易にする機能
- 自動化 Tool : **JUNOScript / Event Policy**
 - 運用を自動化するユーティリティ
- **Etc...**

JUNOScript の概要

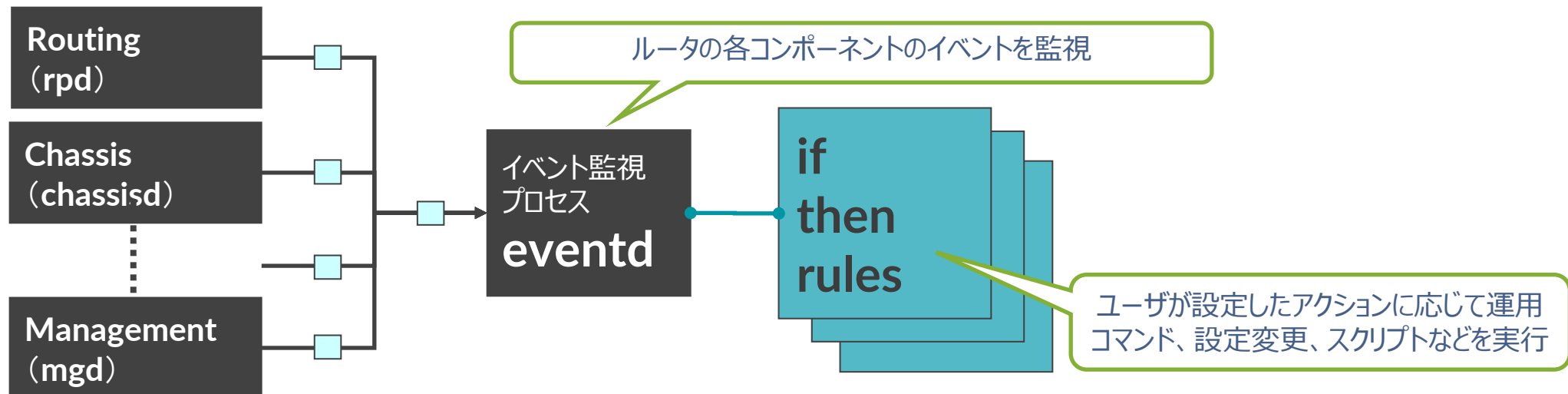
- JUNOScript とは Juniper のネットワーク装置上で動作させることができるスクリプティング機能
- Junos 自体に手を加える必要がないため、Junos の安定性を損なうことなく、ユーザ個別の自動化に対する要望に対し柔軟かつ速やかに対応することが可能
- 大別すると、運用者が起動するスクリプト “Commit Script”、“Op Script” とシステムが起動するスクリプト “Event Policy”、“Event Script” が存在

XSLT / SLAXベースのスクリプト



Junos: Event Policy/Script

- ネットワーク機器上のイベントやタイマーをトリガーとして、コマンドやスクリプトを実行することで、運用の自動化が可能
 - イベントをトリガーとしたアクションを実行 (**Self-monitor**)
 - ルータ上の特定のイベントをトリガーとして、コマンドやスクリプトを実行
 - タイマーをトリガーとしたアクションの実行
 - インターバル設定や日時指定に応じて、コマンドやスクリプトを実行

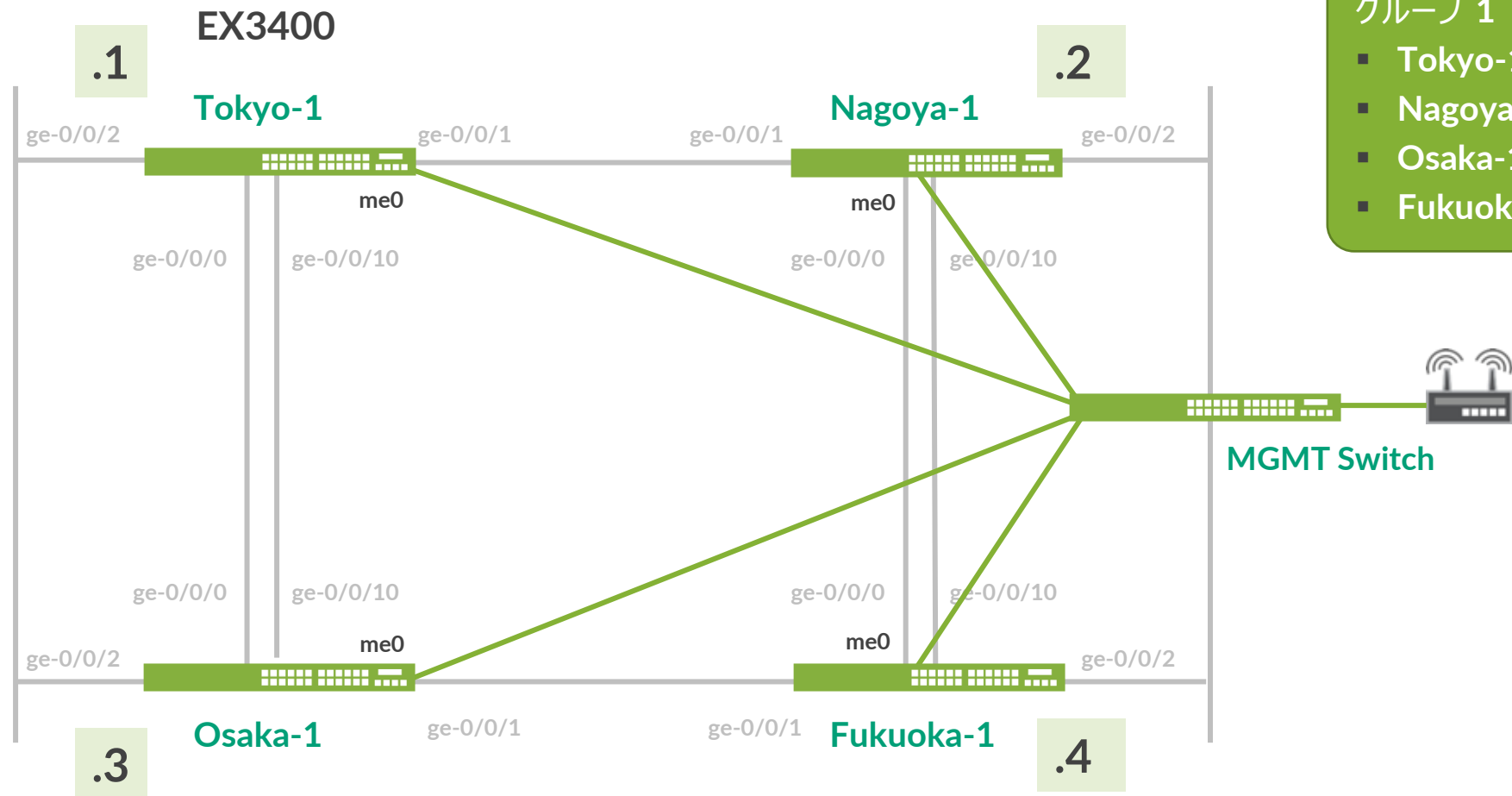




トレーニング・デバイスへの アクセス方法

Ethernet Switching “EX/QFX” Course

Topology – グループ 1

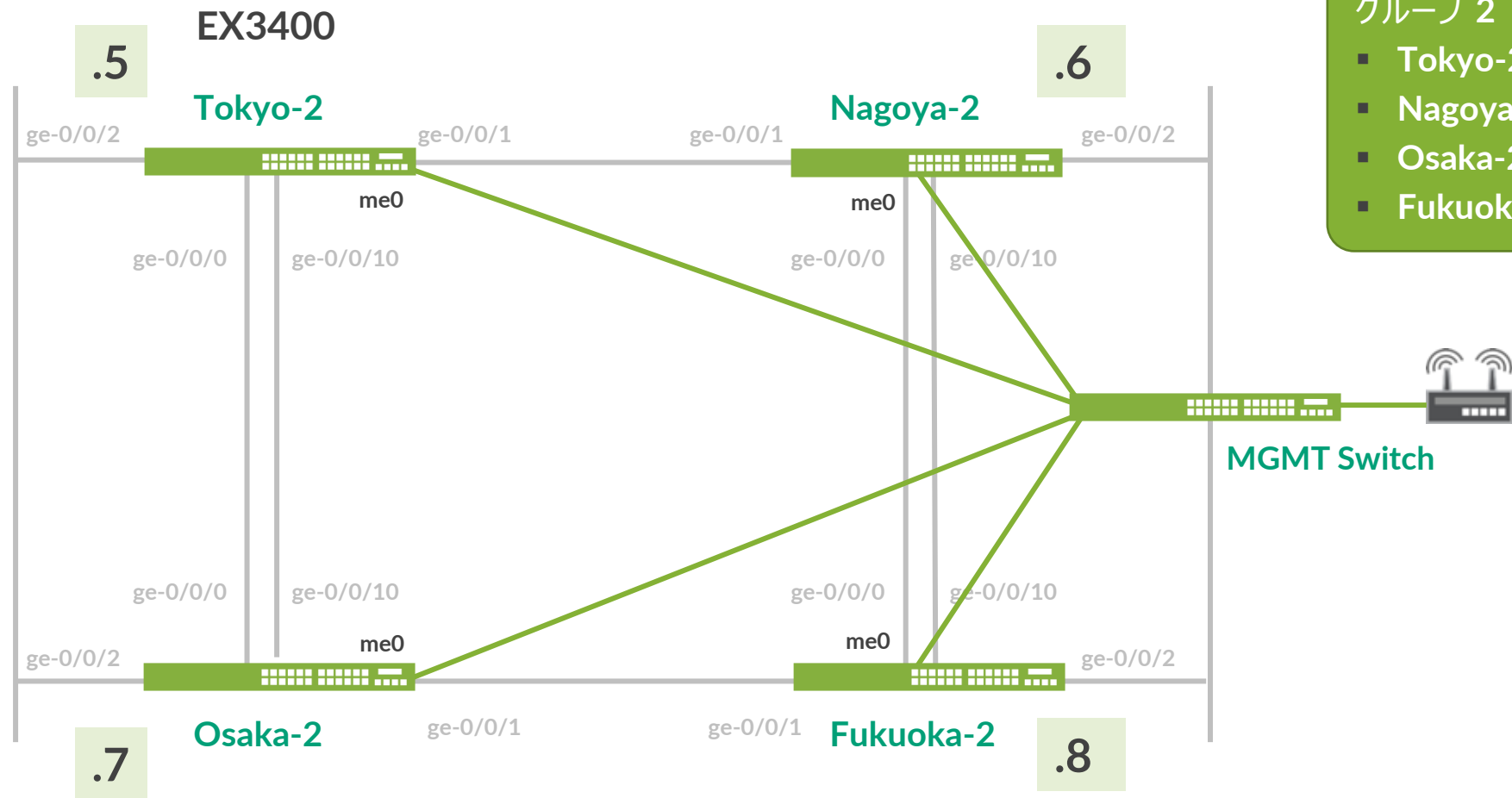


管理用 IP :
(me0) 192.168.1.x/24

- グループ 1
- Tokyo-1 : .1
 - Nagoya-1 : .2
 - Osaka-1 : .3
 - Fukuoka-1 : .4

Ethernet Switching “EX/QFX” Course

Topology – グループ 2



管理用 IP :
(me0) 192.168.1.x/24

- グループ 2
- Tokyo-2 : .5
 - Nagoya-2 : .6
 - Osaka-2 : .7
 - Fukuoka-2 : .8

EX へのログイン

- 初期設定状態の EX にアカウント “root” でログイン
- CLI コマンドで Junos の Operational モードを起動
 - root アカウントは Serial Console、または SSH 接続のみ使用可能
 - 今回は事前に IP アドレス、root パスワード、SSH サービスが設定済みの状態
 - Tera Term から SSHv2 接続で接続してください

接続詳細	
IP アドレス :	192.168.1.x
サービス :	SSH (Tera Term)
ユーザ名 :	root
パスワード :	Juniper

```
--- JUNOS 20.2R3-S2.5 built 2021-07-30 09:45:37 UTC
root% cli
root>
```

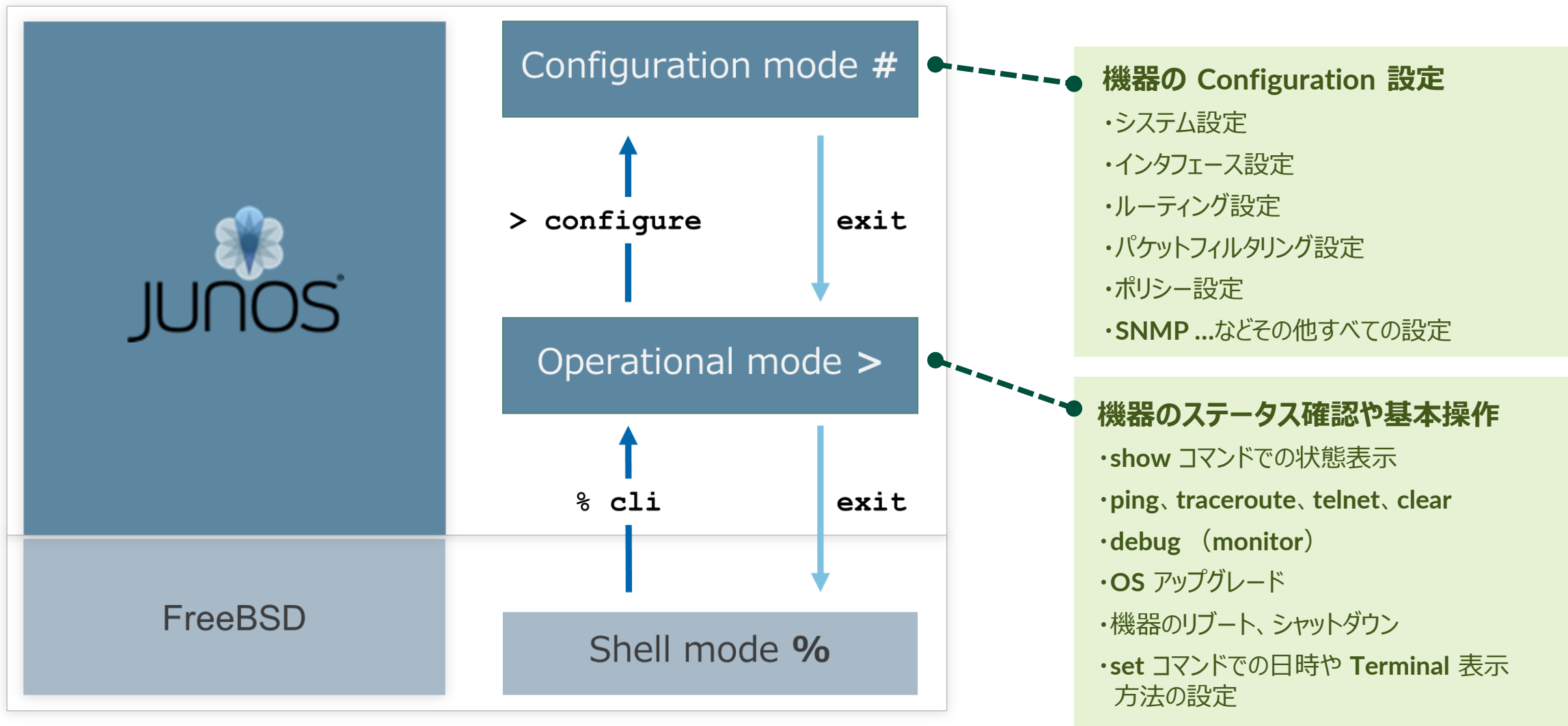




CLI モードと各モード間の移動

CLI 概要

- Junos CLI の 3 つのモード遷移について



Operational モード

- **root ユーザで Login** すると **Shell モード**（プロンプトが “%” ）にアクセス
 - “cli” と投入することで **Shell モード**から **Operational モード**へ移動

```
login: root
Password:

--- JUNOS 20.2R3-S2.5 built 2021-07-30 09:45:37 UTC
root%
root% cli
root@srx>
```

- **root ユーザ以外で Login** すると、**Operational モード**（プロンプトが “>” ）にアクセス
 - “start shell” と投入することで **Operational モード**から **Shell モード**へ移動

```
login: user
Password:

--- JUNOS 20.2R3-S2.5 built 2021-07-30 09:45:37 UTC
user>
user> start shell
%
```

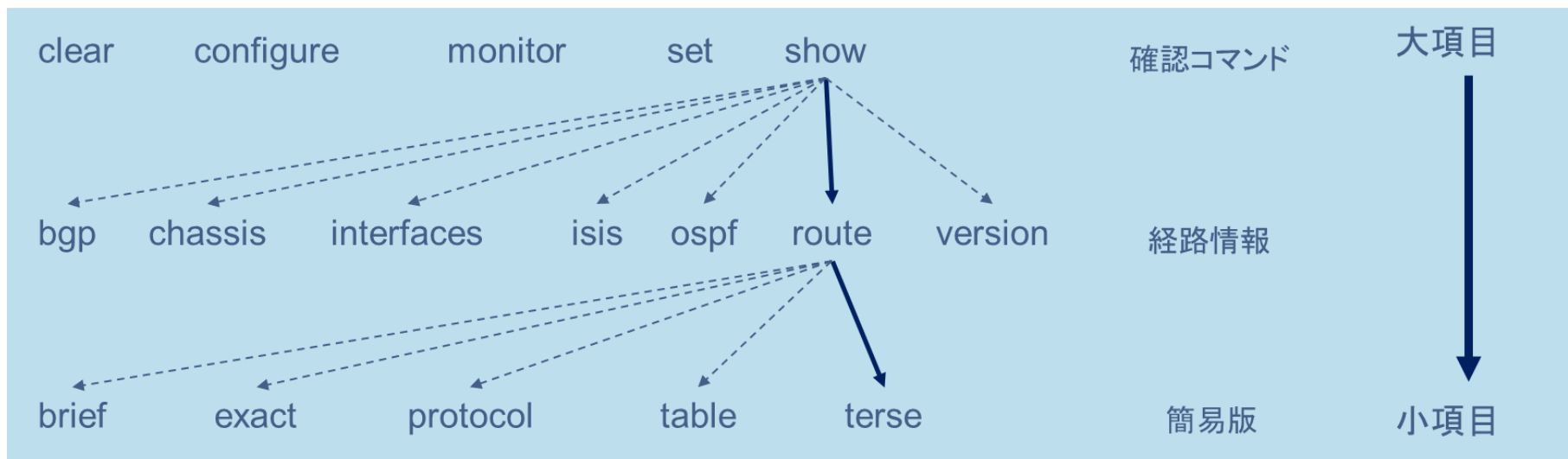
Operational モード

- Operational モードではステータスの確認やシステム操作などのコマンドを提供

```
user> ?
Possible completions:
clear          Clear PPM related statistics information
configure     Manipulate software configuration information
file          Perform file operations
help          Provide help information
load          Load information from file
monitor       Show real-time debugging information
mtrace        Trace multicast path from source to receiver
op            Invoke an operation script
ping          Ping remote target
quit          Exit the management session
request       Make system-level requests
restart       Restart software process
scp           Copy files via ssh
set           Set CLI properties, date/time, craft interface message
show          Show system information
ssh           Start secure shell on another host
start         Start shell
telnet        Telnet to another host
test          Perform diagnostic debugging
traceroute    Trace route to remote
```

Operational モード

- コマンドは階層構造で構成
 - 例：経路情報（簡易版）を確認



```
user> show route terse
```

```
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A	V	Destination	P	Prf	Metric 1	Metric 2	Next hop	AS path
*	?	0.0.0.0/0	S	5			>192.168.1.254	
*	?	192.168.1.0/24	D	0			>ge-0/0/0.0	
*	?	192.168.1.1/32	L	0			Local	

Configuration モード

- Operational モードにて **configure** と投入することで **Configuration モードへ移動**

```
user> configure
Entering configuration mode

[edit]
user#
```

- 他のユーザが **Configuration モードにアクセス中は以下のように表示**

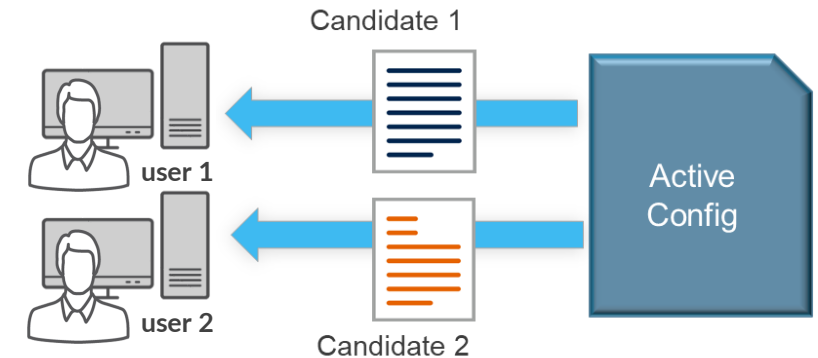
```
user> configure
Entering configuration mode
Users currently editing the configuration:
  user terminal u0 (pid 6898) on since 2022-07-15 09:15:04 UTC, idle 00:05:48
  commit-at
The configuration has been changed but not committed

[edit]
user#
```

Configuration モード： オプション

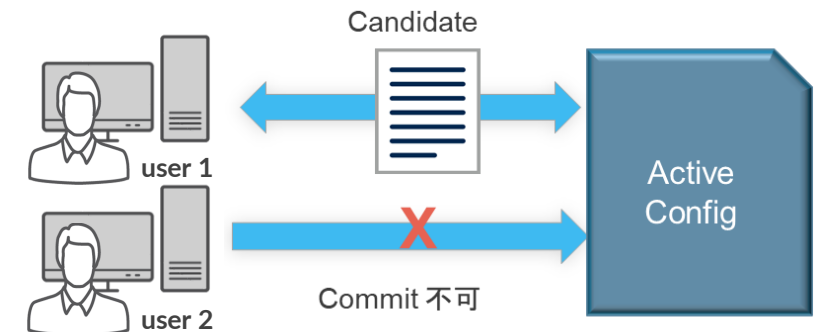
- **configure private** コマンドを使用すると、ログインユーザー専用の **Candidate Configuration** が用意される

```
user> configure private  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```



- **configure exclusive** コマンドを使用すると、ログインユーザーが設定変更を行っている最中に他のログインユーザーが設定変更を行うことを禁止することが可能

```
user> configure exclusive  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```





Junos CLI 操作 ～ Operational モード ～

show コマンド

- **show** コマンド： システム、ステータスに関する情報を表示

> show arp	: ARP テーブルの表示
> show chassis environment	: 温度、ファンなどの環境状態の表示
> show chassis hardware	: ハードウェア情報（シリアルナンバー等）の表示
> show chassis routing-engine	: ルーティングエンジン（CPU や Memory）の状態の表示
> show configuration	: 稼働中の設定の表示
> show interfaces	: Interface の状態の表示
> show route	: 経路情報の表示
> show system uptime	: 稼働時間の表示
> show system users	: ユーザのログイン状況の表示
> show system alarms	: システムアラームの有無の表示
> show version	: Junos ソフトウェアバージョンの表示

show コマンド： オプション

- **show** コマンドでは **terse**、**brief**、**detail**、もしくは **extensive** オプションを使用することで確認できる情報量が指定可能
- **terse**、**brief** のオプションはオプションなしの出力結果と比べ、より簡易的な情報が表示される
- **detail**、**extensive** のオプションはオプションなしの際と比べ、より詳細な情報が表示される

※ コンソールの便利機能 （別途「**Configuration** モード」パートで詳しく説明）

- ショートカットキー： カーソル操作、コマンド履歴、など
- 補完機能： **Space**、**Tab** キー
- 構文チェック

show コマンド : オプション

> show interfaces ge-0/0/0 terse

```
user> show interfaces ge-0/0/0 terse
Interface           Admin Link Proto   Local            Remote
ge-0/0/0            up    up
ge-0/0/0.0          up    up   inet    192.168.1.1/24
```

> show interfaces ge-0/0/0 brief

```
user> show interfaces ge-0/0/0 brief
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 1000mbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
  Auto-negotiation: Enabled, Remote fault: Online
Device flags      : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags        : None

Logical interface ge-0/0/0.0
  Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
  Security: Zone: Null
  inet 192.168.1.1/24
```

show コマンド : オプション

> show interfaces ge-0/0/0 (オプションなし)

```
user> show interfaces ge-0/0/0
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 138, SNMP ifIndex: 513
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: ec:13:db:db:65:80, Hardware address: ec:13:db:db:65:80
  Last flapped   : 2022-08-01 16:07:41 UTC (00:09:59 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics
    Bit errors           Seconds
    Errored blocks       0
  Ethernet FEC statistics
    FEC Corrected Errors      0
    FEC Uncorrected Errors    0
    FEC Corrected Errors Rate 0
    FEC Uncorrected Errors Rate 0
  Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 72) (SNMP ifIndex 521)
  Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
  Input packets : 0
  Output packets: 27
  Security: Zone: Null
  Protocol inet, MTU: 1500
  Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 1,
  Curr new hold cnt: 1, NH drop cnt: 0
  Flags: Sendbcst-pkt-to-re, Is-Primary
  Addresses, Flags: Is-Default Is-Preferred Is-Primary
    Destination: 192.168.1/24, Local: 192.168.1.1, Broadcast: 192.168.1.255
```

show コマンド : オプション

> show interfaces ge-0/0/0 detail

```
user> show interfaces ge-0/0/0 detail
Physical interface: ge-0/0/0, Enabled, Physical link is Up
Interface index: 138, SNMP ifIndex: 513, Generation: 141
Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000Mbps, BFDU Error: None, Loop Detect PDU Error: None,
Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Disabled, Auto-negotiation: Enabled,
Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags : None
CoS queues : 8 supported, 8 maximum usable queues
Hold-times : Up 0 ms, Down 0 ms
Current address: ec:13:db:db:65:80, Hardware address: ec:13:db:db:65:80
Last flapped : 2022-08-01 16:07:41 UTC (00:11:32 ago)
Statistics last cleared: Never
Traffic statistics:
  Input bytes : 0 0 bps
  Output bytes : 1386 0 bps
  Input packets: 0 0 pps
  Output packets: 33 0 pps
Egress queues: 8 supported, 4 in use
Queue counters: Queued packets Transmitted packets Dropped packets
  0 32 32 0
  1 0 0 0
  2 0 0 0
  3 0 0 0
Queue number: Mapped forwarding classes
  0 best-effort
  1 expedited-forwarding
  2 assured-forwarding
  3 network-control
Active alarms : None
Active defects : None
PCS statistics
  Bit errors 0
  Errored blocks 0
Ethernet FEC statistics
  FEC Corrected Errors 0
  FEC Uncorrected Errors 0
  FEC Corrected Errors Rate 0
  FEC Uncorrected Errors Rate 0
Interface transmit statistics: Disabled
MACSec statistics:
  Output
    Secure Channel Transmitted
      Protected Packets : 0
      Encrypted Packets : 0
      Protected Bytes : 0
      Encrypted Bytes : 0
  Input
    Secure Channel Received
      Accepted Packets : 0
      Validated Bytes : 0
      Decrypted Bytes : 0
Logical interface ge-0/0/0.0 (Index 72) (SNMP ifIndex 521) (Generation 142)
Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
Traffic statistics:
  Input bytes : 0
  Output bytes : 1386
  Input packets: 0
```

```
Output packets: 33
Local statistics:
  Input bytes : 0
  Output bytes : 1386
  Input packets: 0
  Output packets: 33
Transit statistics:
  Input bytes : 0 0 bps
  Output bytes : 0 0 bps
  Input packets: 0 0 pps
  Output packets: 0 0 pps
Security: Zone: Null
Flow Statistics :
Flow Input statistics :
  Self packets : 0
  ICMP packets : 0
  VPN packets : 0
  Multicast packets : 0
  Bytes permitted by policy : 0
  Connections established : 0
Flow Output statistics:
  Multicast packets : 0
  Bytes permitted by policy : 0
Flow error statistics (Packets dropped due to):
  Address spoofing: 0
  Authentication failed: 0
  Incoming NAT errors: 0
  Invalid zone received packet: 0
  Multiple user authentications: 0
  Multiple incoming NAT: 0
  No parent for a gate: 0
  No one interested in self packets: 0
  No minor session: 0
  No more sessions: 0
  No NAT gate: 0
  No route present: 0
  No SA for incoming SPI: 0
  No tunnel found: 0
  No session for a gate: 0
  No zone or NULL zone binding: 0
  Policy denied: 0
  Security association not active: 0
  TCP sequence number out of window: 0
  Syn-attack protection: 0
  User authentication errors: 0
Protocol inet, MTU: 1500
Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 1,
Curr new hold cnt: 1, NH drop cnt: 0
Generation: 156, Route table: 0
Flags: Sendbroadcast-pkt-to-re, Is-Primary
Addresses, Flags: Is-Default Is-Preferred Is-Primary
Destination: 192.168.1/24, Local: 192.168.1.1, Broadcast: 192.168.1.255,
Generation: 156
```

show コマンド : オプション

> show interfaces ge-0/0/0 extensive

```
user> show interfaces ge-0/0/0 extensive
Physical interface: ge-0/0/0, Enabled, Physical link is Up
Interface index: 138, SNMP ifIndex: 513, Generation: 141
Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BFDU Error: None, Loop Detect PDU Error: None,
Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Disabled, Auto-negotiation: Enabled,
Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags : None
CoS queues : 8 supported, 8 maximum usable queues
Hold-times : Up 0 ms, Down 0 ms
Current address: ec:13:db:65:80, Hardware address: ec:13:db:65:80
Last flapped : 2022-08-01 16:07:41 UTC (00:12:51 ago)
Statistics last cleared: Never
Traffic statistics:
  Input bytes : 0 0 bps
  Output bytes : 1596 0 bps
  Input packets: 0 0 pps
  Output packets: 38 0 pps
Dropped traffic statistics due to STP State:
  Input bytes : 0
  Output bytes : 0
  Input packets: 0
  Output packets: 0
Input errors:
  Errors: 0, Drops: 0, Framing errors: 0, Runt: 0, Policed discards: 0,
  L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
  FIFO errors: 0, Resource errors: 0
Output errors:
  Carrier transitions: 1, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
  FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Egress queues: 8 supported, 4 in use
Queue counters: Queued packets Transmitted packets Dropped packets
  0 37 37 0
  1 0 0 0
  2 0 0 0
  3 0 0 0
Queue number: Mapped forwarding classes
  0 best-effort
  1 expedited-forwarding
  2 assured-forwarding
  3 network-control
Active alarms : None
Active defects : None
PCS statistics
  Bit errors 0
  Errored blocks 0
Ethernet FEC statistics
  FEC Corrected Errors 0
  FEC Uncorrected Errors 0
  FEC Corrected Errors Rate 0
  FEC Uncorrected Errors Rate 0
MAC statistics:
  Receive Transmit
  Total octets 0 2368
  Total packets 0 37
  Unicast packets 0 0
  Broadcast packets 0 37
  Multicast packets 0 0
  CRC/Align errors 0 0
  FIFO errors 0 0
  MAC control frames 0 0
```

```
MAC pause frames 0 0
Oversized frames 0
Jabber frames 0
Fragment frames 0
VLAN tagged frames 0
Code violations 0
Filter statistics:
  Input packet count 0
  Input packet rejects 0
  Input DA rejects 0
  Input SA rejects 0
  Output packet count 0
  Output packet pad count 0
  Output packet error count 0
CAM destination filters: 2, CAM source filters: 0
Autonegotiation information:
  Negotiation status: Complete
  Link partner:
    Link mode: Full-duplex, Flow control: None, Remote fault: OK
Local resolution:
  Flow control: None, Remote fault: Link OK
Packet Forwarding Engine configuration:
  Destination slot: 0
CoS information:
  Direction : Output
  CoS transmit queue
    % bps % Buffer Priority Limit
    0 best-effort 95 950000000 95 0 low none
    3 network-control 5 500000000 5 0 low none
Interface transmit statistics: Disabled
MACSec statistics:
  Output
    Secure Channel Transmitted
      Protected Packets : 0
      Encrypted Packets : 0
      Protected Bytes : 0
      Encrypted Bytes : 0
  Input
    Secure Channel Received
      Accepted Packets : 0
      Validated Bytes : 0
      Decrypted Bytes : 0
Logical interface ge-0/0/0.0 (Index 72) (SNMP ifIndex 521) (Generation 142)
Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
Traffic statistics:
  Input bytes : 0
  Output bytes : 1596
  Input packets: 0
  Output packets: 38
Local statistics:
  Input bytes : 0
  Output bytes : 1596
  Input packets: 0
  Output packets: 38
Transit statistics:
  Input bytes : 0 0 bps
  Output bytes : 0 0 bps
  Input packets: 0 0 pps
  Output packets: 0 0 pps
Security: Zone: Null
Flow Statistics :
Flow Input statistics :
```

```
Self packets : 0
ICMP packets : 0
VPN packets : 0
Multicast packets : 0
Bytes permitted by policy : 0
Connections established : 0
Flow Output statistics:
  Multicast packets : 0
  Bytes permitted by policy : 0
Flow error statistics (Packets dropped due to):
  Address spoofing: 0
  Authentication failed: 0
  Incoming NAT errors: 0
  Invalid zone received packet: 0
  Multiple user authentications: 0
  Multiple incoming NAT: 0
  No parent for a gate: 0
  No one interested in self packets: 0
  No minor session: 0
  No more sessions: 0
  No NAT gate: 0
  No route present: 0
  No SA for incoming SPI: 0
  No tunnel found: 0
  No session for a gate: 0
  No zone or NULL zone binding: 0
  Policy denied: 0
  Security association not active: 0
  TCP sequence number out of window: 0
  Syn-attack protection: 0
  User authentication errors: 0
Protocol inet, MTU: 1500
Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 1,
Curr new hold cnt: 1, NH drop cnt: 0
Generation: 156, Route table: 0
Flags: Sendbcst-pkt-to-re, Is-Primary
Addresses, Flags: Is-Default Is-Preferred Is-Primary
  Destination: 192.168.1/24, Local: 192.168.1.1, Broadcast: 192.168.1.255,
  Generation: 156
```

コンソール画面出力に関する操作

- 画面に **---(more)---** prompt が表示されているときは以下のキーを使用して操作が可能

Space:	次画面に進む
b:	前画面に戻る
d:	½ 画面進む
Enter:	1 行進む
/string:	検索
n:	再検索
q:	プロンプトに戻る（出力の Abort ）
h:	これらキーヘルプの表示

```
user> show configuration
## Last commit: 2022-07-15 10:04:45 UTC by user
version 20.2R3-S2.5;
system {
    root-authentication {
        encrypted-password
"$6$zD7ag5vO$7IFu12bzwmnRtLm4OE9546HZ6Dgkty6wfaYefYRqgd1AI
Pus0hghi6IuBPvMfdT.CxNQFuzSqbeQO86HpiZbv0"; ## SECRET-DATA
    }
    login {
        user user {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password
"$6$SZPCL2gd$UICNYS6sUhKvfDVWg9.hkm9r0H1QZu1rpSzUa9VgfyEFF
ez1N4/1w17Dy6N0wFX0iLJvZ7/wqPYS7ZP.ETgYb1"; ## SECRET-DATA
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
---(more)---
```


コンソール画面出力に関する操作 | no-more

- コンソール出力は **CLI** のスクリーンサイズを考慮して動作
- 出力内容が多い場合、**CLI** 画面に **---(more)---** を表示し、出力を分けて表示
- “ | **no-more**” オプションを使用し、出力全体を一度に表示することが可能

```
user> show configuration | no-more
## Last commit: 2022-07-15 10:04:45 UTC by user
version 20.2R3-S2.5;
system {
    root-authentication {
        encrypted-password
"$6$zD7ag5vO$7IFu12bzwmnRtLm4OE9546HZ6Dgkty6wfaYefYRqgd1AIPus0hghi6IuBPvMfdT.CxNQFuzSqbeEQ086HpiZb
v0"; ## SECRET-DATA
    }
    login {
        user user {
            uid 2000;
            :
            :
```

※ “ **set cli screen-length < 行数 >** ” コマンドで **more** 表示の行数指定も可能

パイプ “|” オプションの利用

- Unix 同様のパイプ “|” をサポート、**config** や **show** コマンドなどにて有効利用
 - root@lab> show configuration | display set
 - root@lab> show log messages | no-more
 - root@lab> show route | find 192.168.1.0
 - root@lab# show interface | save interface_config.txt

```
user> show configuration | ?
Possible completions:
  append      Append output text to file
  compare    Compare configuration changes with prior version
  count       Count occurrences
  display    Show additional kinds of information
  except      Show only text that does not match a pattern
  find       Search for first occurrence of pattern
  hold        Hold text without exiting the --More-- prompt
  last        Display end of output only
  match      Show only text that matches a pattern
  no-more    Don't paginate output
  request     Make system-level requests
  save       Save output text to file
  tee         Write to standard output and file
  trim        Trim specified number of columns from start of line
```

パイプ “|” 使用例

- **Configuration** の表示方法を変更 (**display set**)
 - 階層表記に加え、行単位での表示も可能

```
user> show configuration interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.1.1/24;
    }
  }
}
irb {
  unit 0 {
    family inet {
      address 192.168.100.1/24;
    }
  }
}
```

```
user> show configuration interfaces | display set
set interfaces ge-0/0/0 unit 0 family inet address
192.168.1.1/24
set interfaces irb unit 0 family inet address
192.168.100.1/24
```



状況に応じ、お好みの表記方法を選択可能

パイプ “|” 使用例

- 稼働中の **Configuration** のファイルへの出力方法

Operational モードにて **show configuration | save** <出力先+ファイル名>

```
user> show configuration | save ftp://test@192.168.1.23/SRX_Config
Password for test@192.168.1.23:
ftp://test@192.168.1.23/SRX_Config          100% of 1356  B 2021 kBps
Wrote 59 lines of output to 'ftp://test@192.168.1.23/SRX_Config'
```

FTP サーバへ出力

- 編集中の **configuration** のファイルへの出力方法

Configuration モードにて **save** <出力先+ファイル名>

```
root# save /config/EDITING-CONFIG.txt
Wrote 59 lines of configuration to '/config/EDITING-CONFIG.txt'
```

/config/ へ出力

※保存先を指定しない場合、**user** の **home directory** へ出力される

- config** の特定の文字列を使用した行の表示 (**match**)

```
user> show configuration | display set | match ssh
set system services ssh root-login allow
set system services ssh protocol-version v2
```

文字列「ssh」を含む行を表示

Junos ファイルシステムの構成について

- Junos では各種構成ファイルや Log ファイルなどをファイルシステム上のディレクトリにて管理される

/config

使用中のコンフィグレーションと過去 3 世代までのコンフィグレーションを格納

/var/db/config

4 世代以降のコンフィグレーションを格納

gz 形式に圧縮されて保存されているが **file show** コマンドで表示可能

FreeBSD では **zcat** コマンドで表示可能

/var/tmp

Junos ソフトウェアアップグレード時など、**image** 格納するディレクトリ

また、各デーモンのコアダンプファイルを格納

/var/log

各種 **Log** や **Trace option** 機能にて取得したデバッグ情報ファイルを格納

/var/home

各ユーザのホームディレクトリが作成される

各ユーザがローカルに保存した情報は全て各ユーザのホームディレクトリに格納

例えば、現在使用中のコンフィグを **save** コマンドにて保存した場合など

Junos ファイルシステムの構成について

- 各ディレクトリに格納しているファイルの確認方法

> **file list** /<directory>/

```
root> file list /var/home/  
/var/home/:  
SAMPLE/
```



/var/home 配下の情報を表示
ユーザ (SAMPLE) のディレクトリが存在

```
root> file list /var/home/SAMPLE/  
/var/home/SAMPLE/:  
TEST-CONFIG
```



/var/home/SAMPLE 配下の情報を表示
ユーザ (SAMPLE) が作成した TEST_CONFIG のファイルが存在

- ディレクトリ配下のファイル内容の確認方法

> **file show** /<directory>/<file_name>

```
root> file show /var/home/SAMPLE/TEST-CONFIG  
## Last changed: 2022-07-15 10:18:41 UTC  
version 20.2R3-S2.5;  
system {  
    root-authentication {  
        encrypted-password  
~~~~~以下省略~~~~~
```



ユーザ (SAMPLE) が作成した
TEST_CONFIG を確認

Junos 運用管理コマンド

- Junos では運用管理に必要な機能をサポート
 - Ping
 - Traceroute
 - Telnet / SSH
 - Monitor

Ping: ネットワークの疎通確認

> ping アドレス + オプション

例 : 172.27.112.1 へ 512 byte の ping を 3 回実施

```
user> ping 192.168.1.23 count 3 size 512
PING 192.168.1.23 (192.168.1.23): 512 data bytes
520 bytes from 192.168.1.23: icmp_seq=0 ttl=128 time=4.446 ms
520 bytes from 192.168.1.23: icmp_seq=1 ttl=128 time=3.995 ms
520 bytes from 192.168.1.23: icmp_seq=2 ttl=128 time=2.633 ms

--- 192.168.1.23 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.633/3.691/4.446/0.771 ms
```

Junos 運用管理コマンド

Traceroute : ネットワークの経路確認

> **traceroute** アドレス + オプション

例 : 8.8.8.8 へ ge-0/0/0 から traceroute を実施

```
user> traceroute 8.8.8.8 interface ge-0/0/0
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 40 byte packets
 1  192.168.1.254 (192.168.1.254)  3.268 ms  1.577 ms  1.349 ms
(snip)
 9  8.8.8.8 (8.8.8.8)  8.195 ms  6.075 ms  5.742 ms
```

Telnet / SSH : ネットワークに接続された機器を操作

> **telnet** アドレス + オプション

例 : 192.168.1.2: port 23 へ telnet を実施

```
user> telnet 192.168.1.2 port 23
Trying 192.168.1.2...
Connected to 192.168.1.2.
Escape character is '^]'.
login:
```

monitor コマンド

- **monitor** コマンド：現在の I/F 別トラフィック状況を表示
 - > **monitor interface traffic**
各 Interface のトラフィックをリアルタイム表示

Interface	Link	Input packets	(pps)	Output packets	(pps)
ge-0/0/0	Up	280	(0)	329	(0)
gr-0/0/0	Up	0	(0)	0	(0)
ip-0/0/0	Up	0	(0)	0	(0)
lsq-0/0/0	Up	0	(0)	0	(0)
lt-0/0/0	Up	0	(0)	0	(0)
mt-0/0/0	Up	0	(0)	0	(0)
sp-0/0/0	Up	0	(0)	0	(0)
ge-0/0/1	Down	0	(0)	0	(0)
ge-0/0/2	Down	0	(0)	0	(0)
ge-0/0/3	Down	0	(0)	0	(0)
ge-0/0/4	Down	0	(0)	0	(0)
ge-0/0/5	Down	0	(0)	0	(0)
ge-0/0/6	Down	0	(0)	0	(0)
ge-0/0/7	Down	0	(0)	0	(0)
esi	Up	0		0	
fti0	Up	0		0	
gre	Up	0		0	
ipip	Up	0		0	
irb	Up	0		0	

Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D

request コマンド

- **request** コマンド： システムの挙動に関するコマンドを実行

※ハンズオン中は **request** コマンドは実施しないようお願いいたします

- システムを再起動

```
> request system reboot
```

- システムをシャットダウン

```
> request system power-off
```

- システムを初期化

```
> request system zeroize
```

- サポートに必要な情報を取得

```
> request support information
```

- 基本となる **Configuration** ファイルを保存（ **rescue config** の保存）

```
> request system configuration rescue save
```

- **OS** をアップグレード

```
> request system software add <ファイル名>
```

Junos のソフトウェアアップグレード

- ソフトウェアアップグレード手順

- 対象の Junos OS をダウンロード

<https://www.juniper.net/support/downloads/group/?f=junos>

- CLI コマンドで Junos ソフトウェアを
FTP/TFTP サーバからデバイス（/var/tmp）に保存

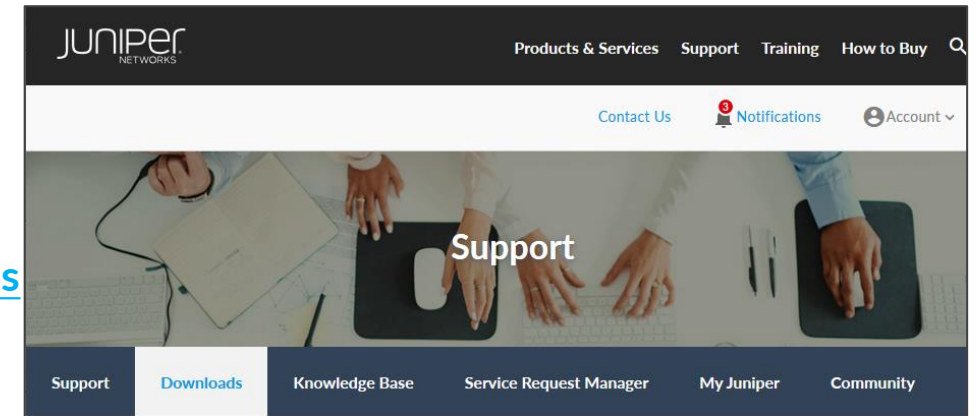
```
> file copy ftp://ログインID @アドレス/Junos パッケージ名/var/tmp
```

- デバイスに保存したパッケージをロード

```
> request system software add /var/tmp/Junos パッケージ名
```

- 機器を再起動

```
> request system reboot
```



```
root> file copy ftp://test@192.168.1.23/junos-  
srxsme-20.2R3-S2.5.tgz /var/tmp  
Password for test@192.168.1.23:  
/var/tmp//...transferring.file.....92LXun/  
100% of 385 MB 2539 kBps 00m00s
```

```
root@> request system software add  
/var/tmp/junos-srxsme-20.2R3-S2.5.tgz  
NOTICE: Validating configuration against  
junos-srxsme-20.2R3-S2.5.tgz.  
(snip)
```

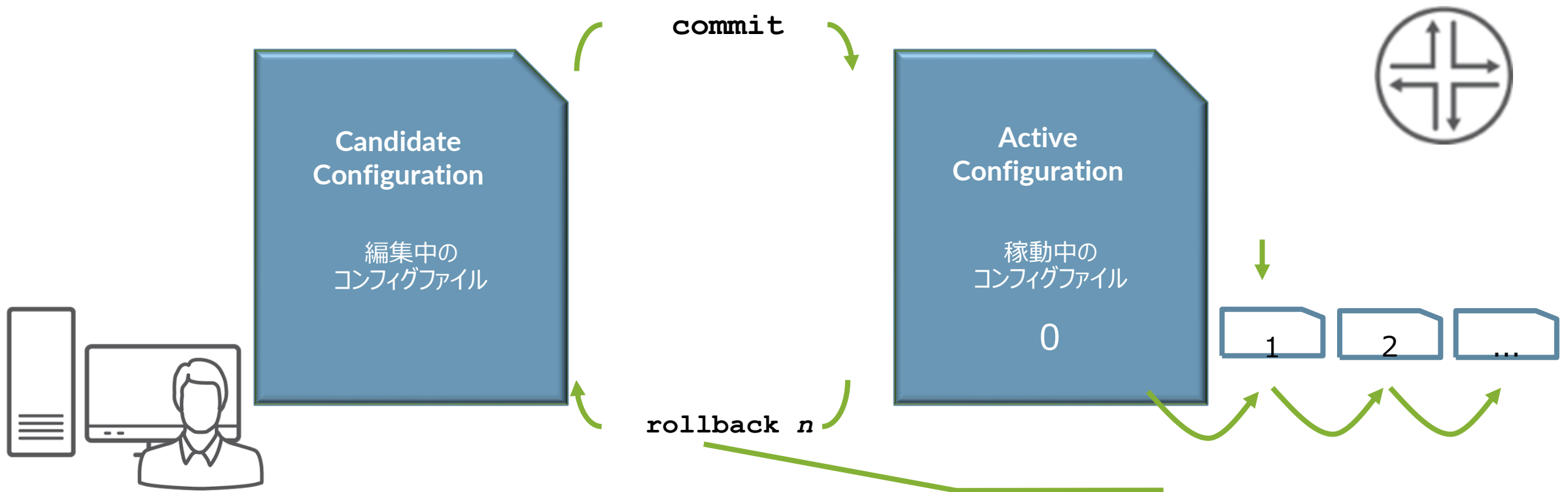
```
root> request system reboot
```




Junos CLI 操作 ～ Configuration モード ～

“Commit & Rollback”

Configuration モードで行った設定変更は、**Candidate Configuration** として保持され、
“commit” するまで設定は **Active Configuration** として反映されない
万一間違えた場合でも、“rollback” コマンドにてすぐに前の状態に戻ることが可能

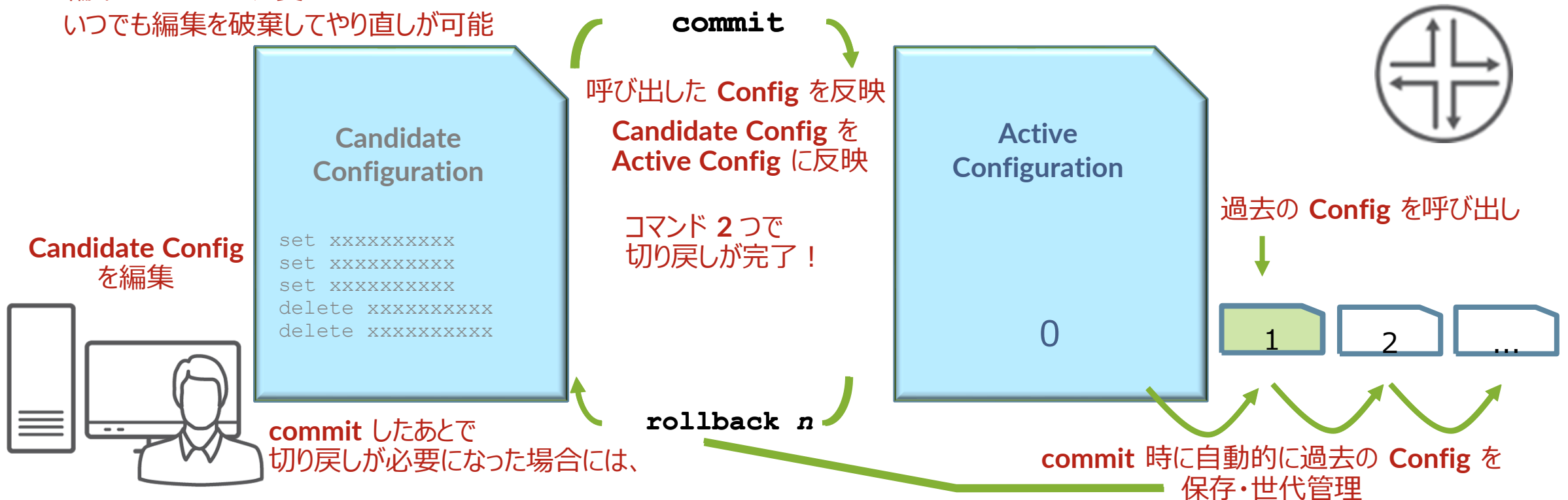


“Commit & Rollback” (アニメ)

Configuration モードで行った設定変更は、**Candidate Configuration** として保持され、
“commit” するまで設定は **Active Configuration** として反映されない
万一間違えた場合でも、“rollback” コマンドにてすぐに前の状態に戻ることが可能

編集したあとに気が変わったら…

いつでも編集を破棄してやり直しが可能

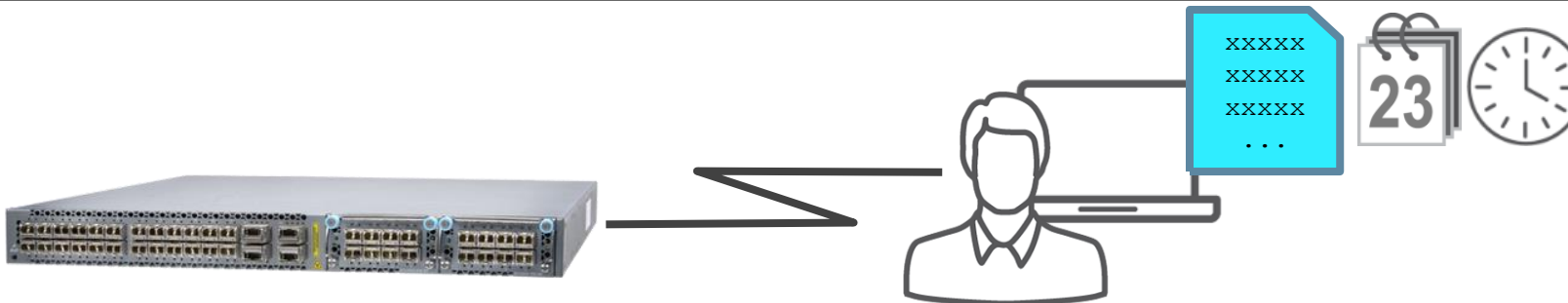


Junos : commit at time オプション

- 設定反映の時間指定（メンテナンスタイムにおける設定反映）
- **commit at xx:xx:xx (time)** コマンドで **commit** すると、指定した時間に設定ファイルを **Activate** することが可能

```
[edit]
root# commit at 10:00:00
configuration check succeeds
commit at will be executed at 2022-07-15 10:00:00 UTC
The configuration has been changed but not committed
Exiting configuration mode
root>
```

メンテナンスタイムに Commit が自動的に実施されるため、
管理者が該当の時間に操作する必要がない



Junos : commit confirmed オプション

- 設定の自動復旧機能（ヒューマンエラーによるトラブル防止のため）
- **commit confirmed** コマンドで **commit** すると、再度 **commit** しない限り **Default 10分** で元の **Config** に **rollback** される
 - 指定した時間あるいは **Default** の **10 分以内** に **2 度目の commit** を入れることで、**Config** は完全に格納される

```
[edit]
root# commit confirmed 5
commit confirmed will be automatically rolled back in 5 minutes unless confirmed
commit complete
# commit confirmed will be rolled back in 5 minutes
```

設定間違いのまま **commit** してしまい **SSH** などが繋がらなくなってしまった後も、一定時間のあと 1 つ前の **Config** に自動復旧するため、リモートデバイスのポリシー変更時などに便利

誤ったアクセスコントロール設定



設定の追加（set）

- **set** コマンド：設定の追加変更
 - **commit** するまでは設定は反映されない

```
user# set system services dns
```

- **commit** することで初めて動作しているデバイスに変更が適用される

```
user# commit  
commit complete
```


設定の削除 (delete)

- **delete** コマンド：設定の削除
 - **commit** するまでは設定は反映されない

```
user# delete system services dns
```

- やはり **commit** することで動作しているデバイスに設定削除の変更が反映される

```
user# commit  
commit complete
```

編集集中の設定確認 (show | compare)

- **show | compare** コマンド：編集集中の設定と稼動中の設定を比較

```
user# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
user# show | compare
[edit]
+ interfaces {
+   ge-0/0/0 {
+     unit 0 {
+       family inet {
+         address 192.168.1.1/24;
+       }
+     }
+   }
+ }
```

Active Config と比較して、ge-0/0/0 に IP アドレスの追加が確認される

+ : 追加

- : 削除

- 過去の Config と編集集中の設定を比較することも可能

```
user# show | compare rollback [1-49]
```

Candidate
Configuration

show | compare

Active
Configuration
(rollback 0)

設定ファイルの復旧 (rollback)

- **rollback** コマンド：設定ファイルの復旧
 - 変更した設定ファイルを破棄する場合に、**rollback** コマンドを投入
(rollback は rollback 0 の略)

```
user# rollback
```

- **rollback n (0-49)** でファイル番号を指定で、過去の設定を **Candidate Config** にコピーすることが可能、容易に過去の状態に戻すことが可能 (過去 50 世代分の設定ファイルを自動保存)

```
user# rollback ?
Possible completions:
  <[Enter]>      Execute this command
  0              2022-07-15 11:12:46 UTC by user via cli
  1              2022-07-15 11:10:41 UTC by user via cli
  2              2022-07-15 11:07:58 UTC by user via cli
  3              2022-07-15 10:18:36 UTC by user via cli
  4              2022-07-15 10:15:12 UTC by user via cli
  5              2022-07-15 10:12:39 UTC by user via cli
  6              2022-07-15 10:04:45 UTC by user via cli
  ... (snip)
```

commit オプション (commit check / at)

- **commit check** コマンド： 構文チェックのみ実行
 - 構文に問題があれば、該当箇所を表示
 - 構文に問題がなくとも **commit** (適用) はされない

```
user# commit check
configuration check succeeds
```

- **commit at** コマンド： 日時を指定して **commit** の実行を予約
 - hh:mm:[ss] または "yyyy-mm-dd hh:mm:[ss]"

```
user# commit at "2022-07-15 11:30"
configuration check succeeds
commit at will be executed at 2022-07-15 11:30:00 JST
Exiting configuration mode
```

Configuration のロード (load)

- **load** コマンド : Configuration ファイルをロード
 - **load** コマンドはいくつかのオプションが存在
 - **load factory-default** 工場出荷時の **Config** をロード
 - **load override <filename>** ロードした **Config** による置き換え
 - **load merge <filename>** ロードした **Config** を追加

```
user# load ?
Possible completions:
factory-default      Override existing configuration with factory default
merge                Merge contents with existing configuration
override             Override existing configuration
patch                Load patch file into configuration
replace              Replace configuration data
set                  Execute set of commands on existing configuration
update              Update existing configuration
```

- **Config** ファイルは外部の **FTP** サーバや機器内ディレクトリからロードすることも可能

```
user# load merge /var/tmp/saved_config.txt
user# load merge ftp://user:passwd@192.168.1.23/saved_config.txt
```

Configuration のロード (load set terminal)

- **load set terminal** コマンド： CLI で追加の **set** コンフィグを貼り付けるときに使用
 - **set** コマンドの大量コピー & ペースト時に **Config** のとりこぼしが防げる

```
user# load set terminal
[Type ^D at a new line to end input]
set services security-intelligence profile feeds-cc-p1 category CC
set services security-intelligence profile feeds-cc-p1 default-rule then action permit
set services security-intelligence profile feeds-cc-p1 default-rule then log
set services security-intelligence profile Inf-hosts category Infected-Hosts
set services security-intelligence profile Inf-hosts default-rule then action permit
set services security-intelligence profile Inf-hosts default-rule then log
set services security-intelligence policy pol-cc CC feeds-cc-p1
set services security-intelligence policy pol-cc Infected-Hosts Inf-hosts
set services advanced-anti-malware policy skyatp_test match application HTTP
set services advanced-anti-malware policy skyatp_test match verdict-threshold 3
set services advanced-anti-malware policy skyatp_test then action permit
set services advanced-anti-malware policy skyatp_test then notification log
set services advanced-anti-malware policy skyatp_test inspection-profile test
set services advanced-anti-malware policy skyatp_test fallback-options action permit
set services advanced-anti-malware policy skyatp_test whitelist-notification log
set services advanced-anti-malware policy skyatp_test blacklist-notification log
load complete
```

< 貼り付け後 **CTRL+D** >

貼り付け対象の
Config を **Terminal**
上でペーストし、最後に
改行してから **CTRL+D**
を押して読み込む

キャンセルしたい場合は
CTRL+C で抜ける

Configuration のロード (load merge terminal)

- **load merge terminal** コマンド： CLI で追加の **Config** を貼り付けるときに使用
 - 大量のコピー&ペースト時にも **Config** のとりこぼしが防げる、最上位の階層から追加の **Config** を投入する階層までのパスが全部必要
 - **relative** オプションを付けると今いる階層に応じて **Config** の階層もショートカットされる

```
[edit]
user# load merge terminal
[Type ^D at a new line to end input]
protocols {
  ospf {
    export static-route;
    area 0.0.0.0 {
      interface ge-0/0/0.0;
      interface ge-0/0/1.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
policy-options {
  policy-statement static-route {
    from {
      protocol static;
      route-filter 10.1.1.0/24 longer;
    }
    then accept;
  }
}
load complete
```

< 貼り付け後 CTRL+D >

Interfaces、protocols
や policy-options など
最上位の構文から記述
していく

```
[edit protocols ospf]
user# load merge terminal relative
[Type ^D at a new line to end input]
area 0.0.0.0 {
  interface ge-0/0/0.0;
}
area 0.0.0.1 {
  stub default-metric 10 no-summaries;
  area-range 192.168.16.0/20;
  interface ge-0/0/3.0;
}
area 0.0.0.2 {
  nssa {
    default-lsa {
      default-metric 20;
      metric-type 1;
      type-7;
    }
    no-summaries;
    area-range 172.16.12.0/22;
  }
  area-range 192.168.48.0/20;
}
load complete
```

< 貼り付け後 CTRL+D >

protocols ospf の階層
に移動し area の
Config だけ追加

protocols { ospf { の
記述は不要

Configuration モード： コマンドサマリー

- 設定&確認コマンド

- **set** : パラメータを設定
- **delete** : パラメータを削除
- **show** : 設定した内容の表示
- **show | compare** : 編集中の **Config** と稼働中の **Config** の差分を表示

- 設定反映コマンド

- **commit** : 編集した設定を **Active Config** に反映
- **rollback** : 過去の **Config** をロードして編集内容を元に戻す
- **load** : 設定したファイルをロード

便利なショートカットキー

- カーソルの移動

Ctrl-B	1 文字戻る
Ctrl-F	1 文字進む
Ctrl-A	行頭に移動
Ctrl-E	行末に移動

- 文字の削除

Delete / Backspace	カーソル前の 1 文字を削除
Ctrl-D	カーソル後の 1 文字を削除
Ctrl-K	カーソルから行末までを削除
Ctrl-U	行をすべて削除
Ctrl-W	現在入力途中の単語または、カーソルより左側の 1 単語を削除

- その他

Ctrl-P or ↑	コマンド履歴の前を表示
Ctrl-N or ↓	コマンド履歴の次を表示
?	次に入力すべきコマンドやパラメータのヒントを表示

コマンド補完と構文エラー

- コマンド補完機能
 - Spaceキー / Tabキー： 固定値を補完
 - Tabキーはユーザが定義した **Policy**名や **Filter**名の補完も可能

```
user# set interfaces ge-0/0/0 unit 0 family inet filter input ?
Possible completions:
  <filter-name>      Name of the filter
  TEST               [firewall filter]

user@srx# set interfaces ge-0/0/0 unit 0 family inet filter input T[tab]
```

- 構文エラーの通知
 - 構文に誤りがあると **syntax error** を表示
 - ^ マークはエラーとなる項目を示す

```
user# load replase
      ^
syntax error, expecting <command>.
```

Configuration モード : Operational モードのコマンドを実行

- **run** コマンドにより、**Configuration** モードにおいて **show** コマンド等を実行し、**status** 等確認することが可能
 - **Operational** モードで確認可能な全てのコマンドの実行が可能
 - **Operational** モードに戻る必要なし

run コマンドを使用し、interface の状態を確認

```
user# run show interfaces
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 138, SNMP ifIndex: 513
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode:
Full-duplex,
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback:
Disabled,
  Source filtering: Disabled, Flow control: Disabled, Auto-
negotiation: Enabled,
  Remote fault: Online
  Device flags      : Present Running

(snip)
```

interface の設定を確認

```
user# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.1.1/24;
    }
  }
}
irb {
  unit 0 {
    family inet {
      address 192.168.100.1/24;
    }
  }
}

(snip)
```



Junos システム設定

システム設定

- **Junos** デバイスのシステムに関する主な設定
 - ユーザ設定
 - ホスト名の設定
 - 時刻設定
 - **DNS** 設定
 - デバイスのサービス設定
 - 管理インタフェース設定
 - ログの設定
 - **SNMP** 設定

システム設定

- ユーザ設定
 - **root** ユーザのパスワードを設定 （※必須設定項目： 未設定の場合 **commit** がエラーとなる）

```
root# set system root-authentication plain-text-password
New password:
Retype new password:
```

- **root** ユーザ以外のユーザアカウントを作成
 - デフォルトでは 3 つのユーザクラスを選択可能
 - **read-only** : **view** （**show** コマンドなど）
 - **operator** : **clear**、**network**、**reset**、**trace**、**view** （デーモンの停止、**ping** / **telnet**、etc）
 - **super-user** : **all** （すべて）

```
root# set system login user TEST class super-user authentication plain-text-password
New password:
Retype new password:
```

システム設定

- ホスト名の設定

```
root# set system host-name LAB
```

- 時刻設定

- Time Zone を指定

```
root# set system time-zone Asia/Tokyo
```

- NTP サーバを指定

```
root# set system ntp server 10.10.10.100
```

- DNS 設定

```
root# set system name-server 192.168.1.100
```

システム設定

- デバイスのサービス設定
 - Telnet、SSH によるアクセスを有効に設定

```
root# set system services telnet
root# set system services ssh
root# set system services ssh root-login allow
```

← root ユーザとして SSH でログインしたい場合に設定

- FTP、Netconf のサービスを有効に設定

```
root# set system services ftp
root# set system services netconf ssh
```

システム設定

- 管理インタフェース設定

- 例 1：EX の管理インタフェース（me0）を設定

```
root# set interfaces me0 unit 0 family inet address 192.168.1.1/24
```

- 例 2：MX、SRX の管理インタフェース（fxp0）を設定

```
root# set interfaces fxp0 unit 0 family inet address 192.168.1.1/24
```

EX3400 rear view



↑
me0

SRX340 front view



↑
fxp0

※管理ポートは、
MX/SRX は “FXP0”、EX は “ME0”、QFX は “EM0”、EX/QFX の VC では “VME（Virtual ME）” と命名
Branch SRX の Low End（SRX300/320）など、Out of Band の管理ポートが無いモデルも存在

システム設定

- ログの設定
 - Syslog サーバ、ファシリティ、ログレベルを指定
 - 例：すべてのレベルのログを **10.10.10.1** へ送信

```
root# set system syslog host 10.10.10.1 any any
```

■ Syslog レベルについて

高	emergency:	ソフトウェアコンポーネントの機能停止を招く状況のメッセージ
	alert:	データベースなどのデータ破損など、直ちに修復が必要な状況のメッセージ
	critical:	物理的なエラーなど重大な問題がある状況のメッセージ
	error:	上記よりも深刻度の低いエラー状況のメッセージ
	warning:	モニタリングの必要性がある状況のメッセージ
	notice:	エラーではないが、特別な処理が必要となる可能性がある状況のメッセージ
	info:	対象のイベントまたは非エラー状況のメッセージ
低	any:	すべてのレベルのメッセージ

システム設定

- SNMP 設定
 - SNMP コミュニティを作成
 - 例：コミュニティ名を **public** に設定、読み込みのみ許可

```
root# set snmp community public authorization read-only
```

- SNMP トラップを設定
 - 例：トラップの送信元を **Loopback 0** に、宛先を **10.10.10.1** に設定

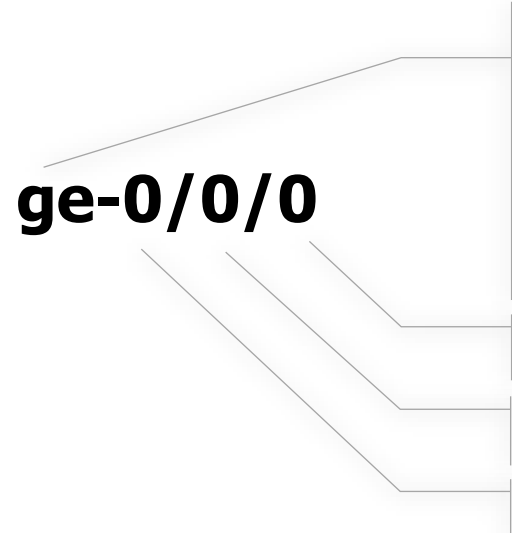
```
root# set snmp trap-options source-address lo0  
root# set snmp trap-group <group-name> targets 10.10.10.1
```



Junos インタフェース設定

インタフェースタイプの表記

- インタフェースタイプは以下のように表記



ge-0/0/0

Type:	fe-x/x/x:	Fast Ethernet ports
	ge-x/x/x:	Gigabit Ethernet ports
	xe-x/x/x:	10 Gigabit Ethernet ports
	et-x/x/x:	40/100 Gigabit Ethernet ports

Port number

PIC slot:	Physical Interface Card → アップリンクモジュール
-----------	---------------------------------------

FPC slot:	Flexible PIC Concentrator (line card) → 筐体ナンバー
-----------	--

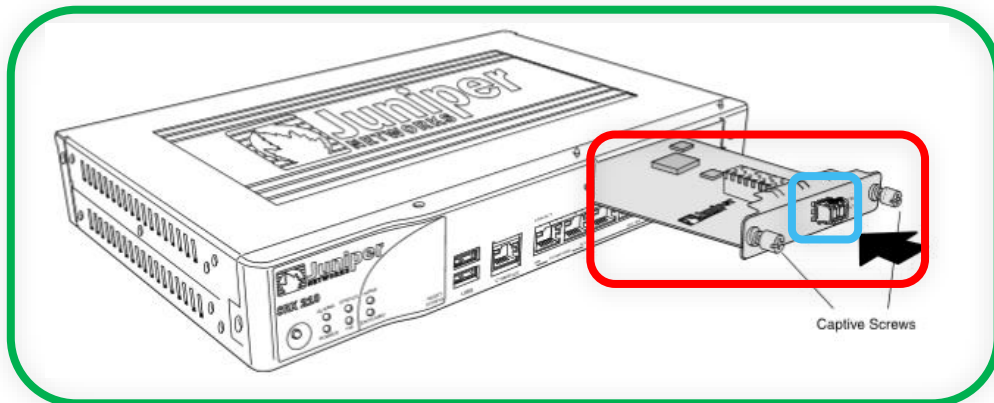
- その他のインタフェース
 - ae : LAG インタフェース
 - lo0 : Loopback インタフェース
 - me0 : EX、QFX シリーズの管理インタフェース
 - fxp0 : SRX、MX シリーズの管理インタフェース

PIC と FPC

- FPC は BOX 型の筐体番号、Chassis 型のラインカード番号に相当
- PIC は FPC に接続されるアップリンクモジュールを指す

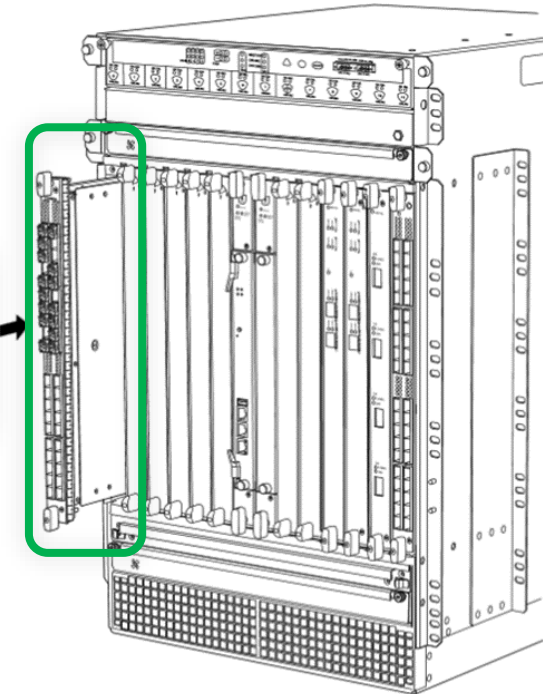
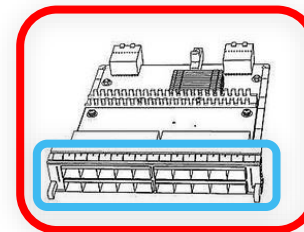
xx - **X** / **X** / **X** FPC PIC Port

BOX 型



※BOX 型における On-Board Port は、xx-**0**/**0**/**X**と表記される

Chassis 型



インタフェース設定

- インタフェースの設定は
物理プロパティの設定と論理プロパティの設定に分けられる
- 物理プロパティの設定
 - データリンクプロトコル
 - リンクスピード、半/全 2 重通信
 - MTU
- 論理プロパティの設定
 - プロトコルファミリー
 - **inet** (IPv4 の設定)
 - **inet6** (IPv6 の設定)
 - **mpls**
 - **ethernet-switching**

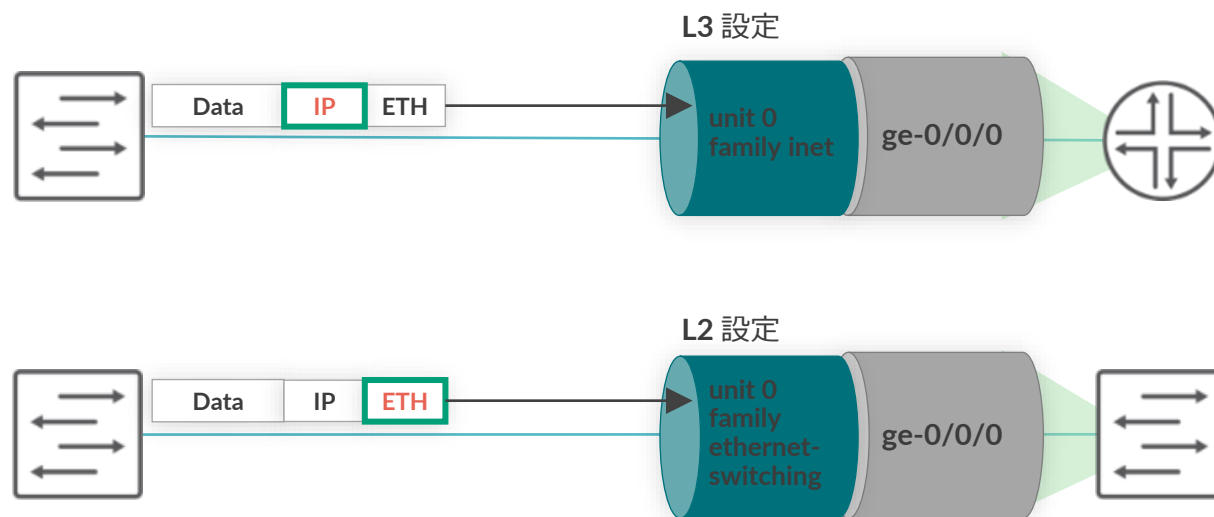
```
interfaces {  
    interface-name {  
        physical-properties;  
        [...]  
        unit unit-number {  
            logical-properties;  
            [...]  
        }  
    }  
}
```

インタフェース名配下に
物理プロパティを設定

unit # 配下に
論理プロパティを設定

Unit ナンバーとは

- ロジカルプロパティを設定するには、“unit” とよばれる単位で設定
 - 一般的なネットワーク OS のサブインタフェースに相当
 - unit 0 はメインインタフェースに相当
 - インタフェースを動作させるためには最低 1 つの unit が必須
 - 1 つの物理インタフェース上に複数の unit を作成することも可能
 - 物理インタフェース ge-0/0/0 の unit 0 は、“ge-0/0/0.0” と表記
 - show コマンドや設定時に unit を指定しなかった場合、自動的に unit 0 として補完

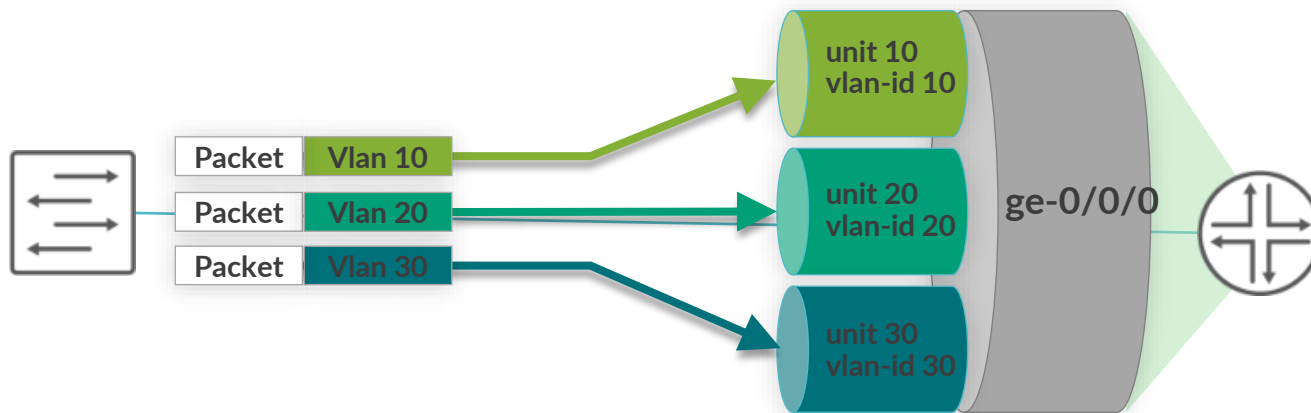


```
ge-0/0/0 {  
  unit 0 {  
    family inet {  
      address 192.168.1.1/24;  
    }  
  }  
}
```

```
ge-0/0/0 {  
  unit 0 {  
    family ethernet-switching {  
      interface-mode access;  
    }  
  }  
}
```

複数 unit の設定例

- 1つの物理インターフェースに複数の **unit** を使用するケース
 - **unit** ごとに **vlan-id** を設定して振り分け
 - **IP アドレス**や **Firewall Filter** も **unit** ごとに個別に設定可能



```
ge-0/0/0 {  
  vlan-tagging;  
  unit 10 {  
    vlan-id 10;  
    family inet {  
      address 192.168.1.1/24;  
    }  
  }  
  unit 20 {  
    vlan-id 20;  
    family inet {  
      address 172.16.1.1/24;  
    }  
  }  
  unit 30 {  
    vlan-id 30;  
    family inet {  
      address 10.1.1.1/24;  
    }  
  }  
}
```

物理 / 論理インタフェース設定例

```
ge-0/0/0 {  
  description TEST;  
  speed 1g;  
  mtu 1400;  
  ether-options {  
    no-auto-negotiation;  
    link-mode full-duplex;  
  }  
}  
-----  
unit 0 {  
  description TEST2;  
  family inet {  
    address 10.10.10.1/24;  
  }  
}  
unit 100 {  
  description TEST3;  
  family inet6 {  
    address 1::1/64;  
  }  
}  
}
```

物理 プロパティ

論理 プロパティ

管理者側から強制的にインタフェースを落とす方法

- **disable** コマンドを使用してインタフェースを落とす（無効化）

```
root# set interfaces ge-0/0/2 disable
```

```
[edit]
```

```
root# commit
```

```
commit complete
```

admin（オペレーター）モードの操作の確認

```
root# show interfaces
```

```
ge-0/0/2 {
```

```
  disable;
```

```
  unit 0 {
```

```
    family inet {
```

```
      address 10.10.10.1/24;
```

admin（オペレータ）の強制的な
インタフェースのダウン

```
root# run show interfaces terse
```

Interface	Admin	Link	Proto
-----------	-------	------	-------

Local	Remote		
-------	--------	--	--

ge-0/0/0	up	up	
----------	----	----	--

ge-0/0/1	up	down	
----------	----	------	--

ge-0/0/2	down	down	
----------	------	------	--

- **disable** コマンドを消去してインタフェースを上げる（有効化）

```
root# delete interfaces ge-0/0/2 disable
```

```
[edit]
```

```
root# commit
```

```
commit complete
```

```
root# run show interfaces terse
```

Interface	Admin	Link	Proto
-----------	-------	------	-------

Local	Remote		
-------	--------	--	--

ge-0/0/0	up	up	
----------	----	----	--

ge-0/0/1	up	down	
----------	----	------	--

ge-0/0/2	up	up	
----------	----	----	--



Junos 経路設定

Static Route の設定

- Static Route 設定

set routing-options static route <あて先アドレス> next-hop <ネクストホップアドレス>

set routing-options static route <あて先アドレス> オプション設定

設定例

```
[edit routing-options]
root# show
static {
  route 0.0.0.0/0 next-hop 172.30.25.1;
  route 172.28.102.0/24 {
    next-hop 10.210.11.190;
    no-readvertise;
  }
}
```

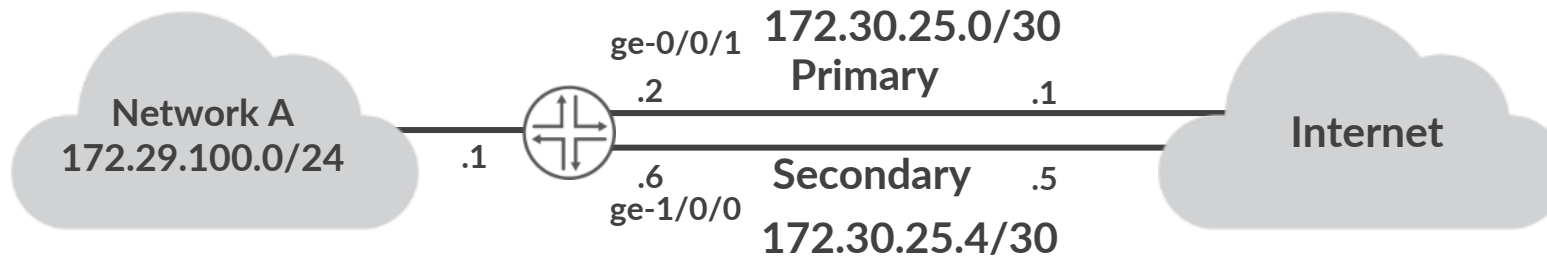
IPv4 デフォルトルートの設定

経路を広報させないための設定
マネージメント用の経路などに利用

制限付きネクストホップの設定

- 同じあて先に **Static Route** を設定する場合は **qualified-next-hop** のオプションを利用し、**preference**（優先）の設定を施す

例：インターネット接続のためのデフォルトルートの設定



```
[edit routing-options]
root# show
static {
  route 0.0.0.0/0 {
    next-hop 172.30.25.1;
    qualified-next-hop 172.30.25.5 {
      preference 7;
    }
  }
}
```

Primary route

※Juniper の static route の preference は 5

Secondary route

※preference を 7 に設定することで優先度を下げる

Static Route の確認

- show コマンドで Static Route を確認

```
root> show route protocol static
```

```
inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
0.0.0.0/0          *[Static/5] 00:00:01  
                  > to 172.30.25.1 via ge-0/0/1.0
```

```
...
```

デフォルトルート

プロトコルと preference

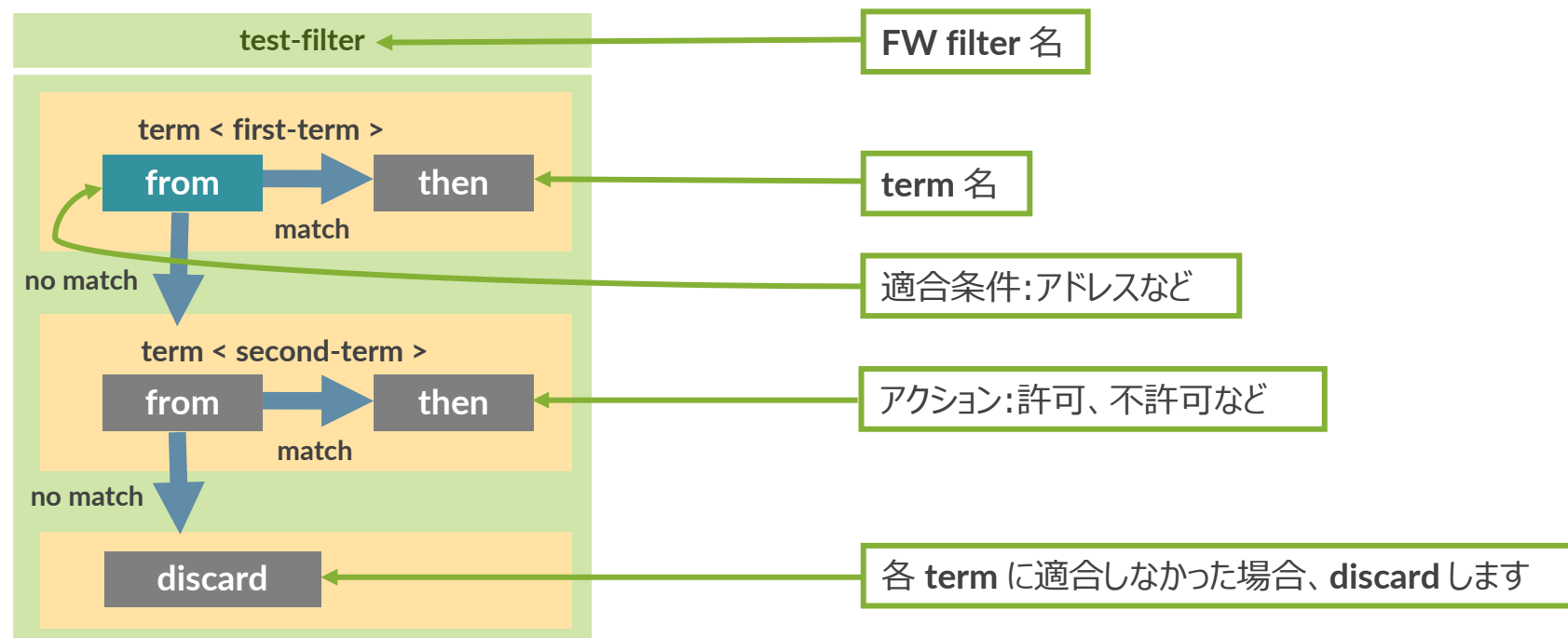
ネクストホップのアドレスとインタフェース



Firewall Filter (ACL) の設定

Firewall Filter の設定

- FW フィルタとは個々のパケットのフローを制御するためのステートレスなフィルタリングポリシー (= **ACL**)
- FW フィルタでは **term** と呼ばれる条件付けのブロックを定義
- フィルタ内の **term** は **top → down** の順番で精査される



※新しく **term** を作成した際など、評価の順番を変更する際は **insert** コマンドを利用して意図した順番に **Term** の入れ替える調整が必要

Firewall Filter の設定

例 1 : 10.10.10.0/24 からの通信を許可しない FW フィルタを作成

```
root# set firewall family inet filter FW-FILTER term BLOCK from source-address 10.10.10.0/24
root# set firewall family inet filter FW-FILTER term BLOCK then discard
root# set firewall family inet filter FW-FILTER term PERMIT then accept
```

```
root# show firewall family inet filter FW-FILTER
```

FW filter 名

```
term BLOCK {
```

term 名

```
  from {
```

```
    source-address {
      10.10.10.0/24;
```

適合条件: 10.10.10.0/24 からの通信

```
    }
```

```
  }
```

```
  then {
```

```
    discard;
```

アクション: 不許可

```
  }
```

```
}
```

```
term PERMIT {
```

```
  then accept;
```

他の IP からの通信を許可

```
}
```


Firewall Filter の設定

例 1：作成した FW フィルタをインタフェースへ適用

```
root# set interfaces ge-0/0/0 unit 0 family inet filter input FW-FILTER
```

```
root# show interfaces ge-0/0/0
unit 0 {
  family inet {
    filter {
      input FW-FILTER;
    }
  }
}
```

← ge-0/0/0 に入ってくる通信に対して FW-FILTER を適用

※ FW フィルタの設定を有効にする際（**commit** する際）に **commit confirm** を利用すると万が一設定を誤ってしまった場合にも切り戻しが可能

Firewall Filter の設定

例 2 : term の順序入れ替え

```
root# set firewall family inet filter FW-FILTER term BLOCK from source-address 10.10.10.0/24
root# set firewall family inet filter FW-FILTER term BLOCK then discard
root# set firewall family inet filter FW-FILTER term PERMIT then accept
root# set firewall family inet filter FW-FILTER term BLOCK2 from protocol udp
root# set firewall family inet filter FW-FILTER term BLOCK2 then discard
```

All permit のあとに term がある
のでこの順序だとこの term は Lookup されない

term は設定した順番で設定ファイルに書き込みが行われる

一方で、意図したフィルターを掛けるためには適切な順序で **term** を記載する必要がある
(上記例では、all PERMIT term の後に BLOCK2 が書かれているので、Lookup がされないことに注意)

- **insert** コマンド : Firewall Filter や Firewall Policy の term 順序を変更

```
root# insert firewall family inet filter FW-FILTER term BLOCK2 before term PERMIT
```

OR

```
root# insert firewall family inet filter FW-FILTER term PERMIT after term BLOCK2
```

Firewall Filter の設定

例 2 : term の順序入れ替え

意図した順番で **term** が記載されていることを確認した上で、**commit** を実行

```
root# show firewall family inet
filter FW-FILTER {
  term BLOCK {
    from {
      source-address {
        10.10.10.0/24;
      }
    }
    then {
      discard;
    }
  }
  term BLOCK2 {
    from {
      protocol udp;
    }
    then {
      discard;
    }
  }
  term PERMIT {
    then accept;
  }
}
```

← insert コマンドにより term BLOCK2 が PERMIT の前に移動している

Firewall Filter の設定

例 3 : Junos 製品へのマネージメント通信を制限

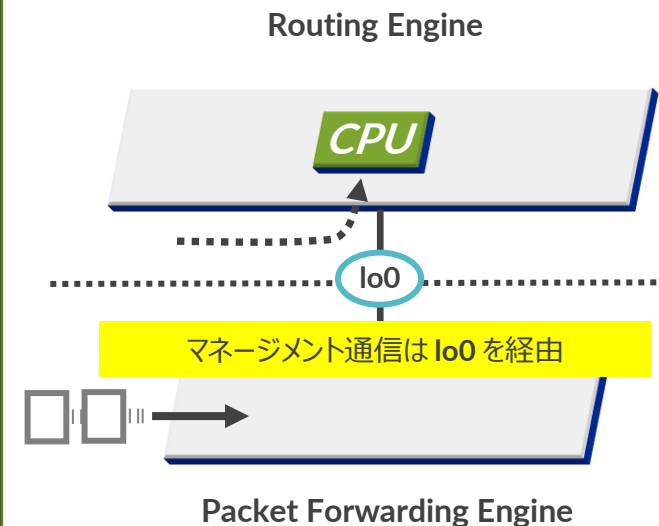
1. FW フィルタを作成

- 192.168.1.0/24 のセグメントから SSH での通信のみ許可

2. 作成した FW フィルタを lo0 （ループバックインタフェース）に適用

```
root# show firewall family inet
filter MANAGEMENT {
  term PERMIT {
    from {
      source-address {
        192.168.1.0/24;
      }
      protocol tcp;
      destination-port ssh;
    }
    then accept;
  }
}
```

```
root# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input MANAGEMENT;
      }
      address 10.10.10.1/24;
    }
  }
}
```



※ EX、QFX シリーズ自身への通信を制御する場合、lo0 および、me0（EX）、em0（QFX）へ Firewall Filter を適用することが必要

※ SRX、MX シリーズ自身への通信を制御する場合、lo0 のみに Firewall Filter を適用することで制御可能
（管理インタフェース fxp0 への適用は不要）



JUNOS Hands On Training “EX / QFX” Course

LAB (後半)

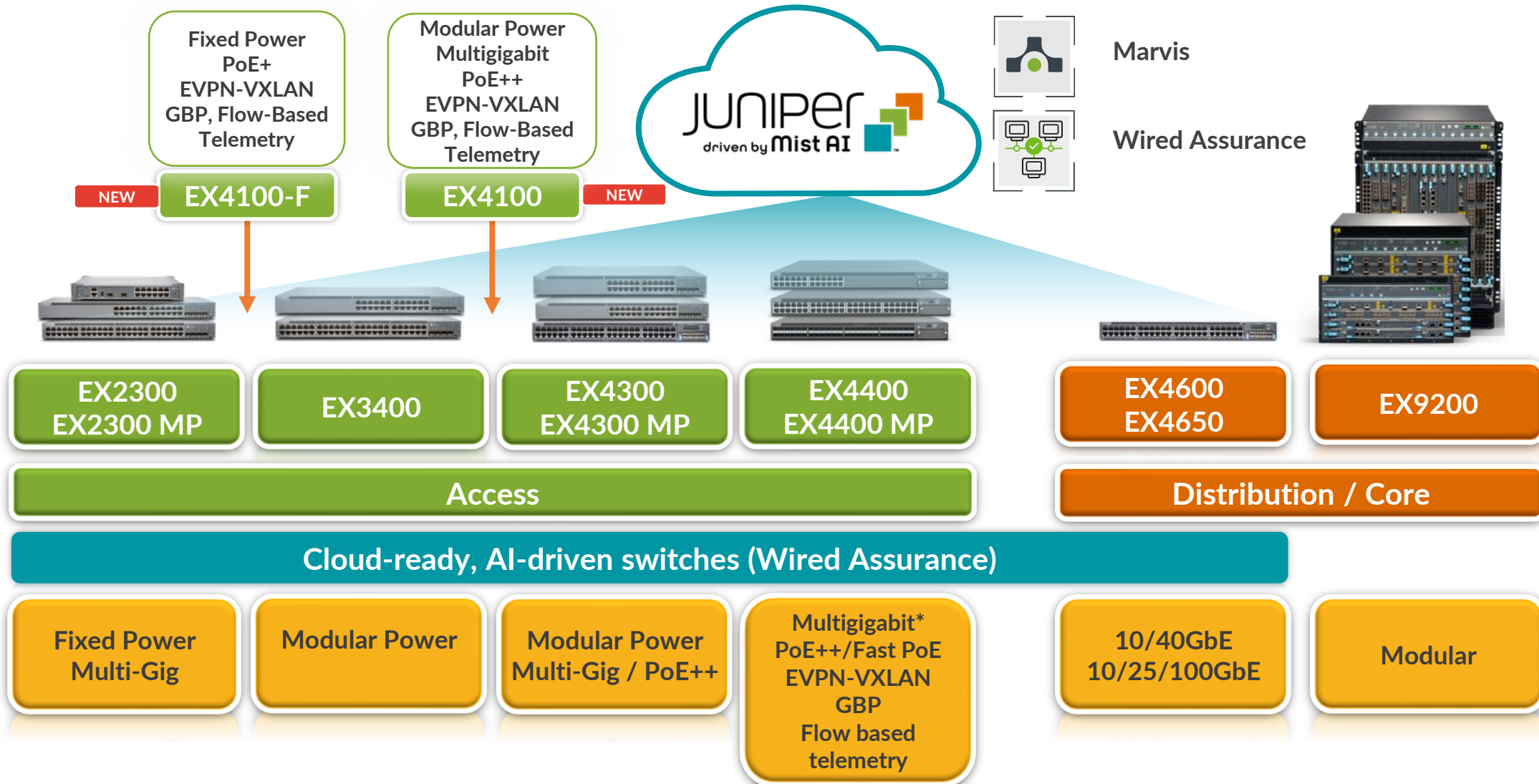
トレーニング概要 「Junos スイッチ “EX / QFX” コース」

トレーニング内容（後半）	記載ページ
Junos EX シリーズ製品紹介	P. 103
LAB.1 Junos の基本的な操作・設定	P. 112
LAB.2 Interface の設定	P. 127
LAB.3 Routing の設定	P. 140
LAB.4 Firewall Filter の設定	P. 147
Virtual Chassis とは	P. 153
Virtual Chassis Deep Dive	P. 165
LAB.5 Virtual Chassis の設定	P. 184
Appendix	P. 202



Juniper EX シリーズ製品紹介

EX シリーズプロダクトポートフォリオ



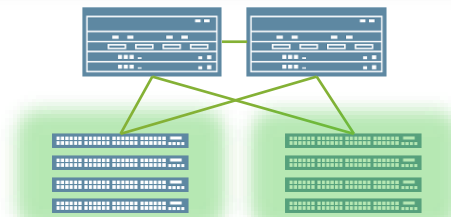
EX シリーズプロダクトポートフォリオ



	EX2300	EX3400	EX4100-F	EX4100	EX4300	EX4400	EX4600	EX4650	EX9200
Positioning	Access mGig Access	Access	Access & Aggregation	mGig Access & Aggregation			Core & Aggregation		Core & Aggregation
Downlink	48 x 1GbE / 16 x mGig & 32 x 1GbE	48 x 1GbE	48 x 1GbE, 24 x 1GbE, 12 x 1GbE	48 x 1GbE, 24 x 1GbE, 16 x mGig & 32 x 1GbE, 8 x mGig & 16 x 1GbE	48 x 1GbE / 24 x mGig & 24 x 1G	48 x 1GbE, 48 x mGig, 12 x SFP+ & 36 x SFP	24 x 10GbE & 4 x 40GbE	48 x 10/25GbE	480 x 10GbE 120 x 40GbE 40 x 100GbE
Uplink	4 x 10GbE / 6 x 10GbE	4 x 1/10GbE & 2 x 40GbE	4 x 10GbE & 4 x 10GbE, 4 x 10GbE & 2 x 10GbE	4 x 25GbE & 4 x 10GbE	10G / 40G / 100G option	10G / 25G option & 2 x 100GbE	8 x 10GbE or 4 x 40GbE uplinks	8 x 40/100GbE uplinks	N/A
PoE	PoE+			POE+(802.3at) POE++ (802.3bt)			N/A		
Fabric	Virtual Chassis		Virtual Chassis EVPN VXLAN		Virtual Chassis	Virtual Chassis EVPN VXLAN			EVPN VXLAN
	Wired Assurance								

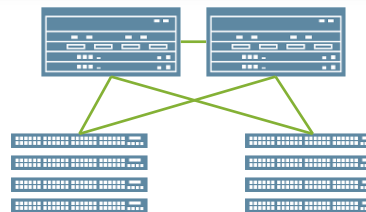
ソリューションとプロダクト構成

Virtual Chassis



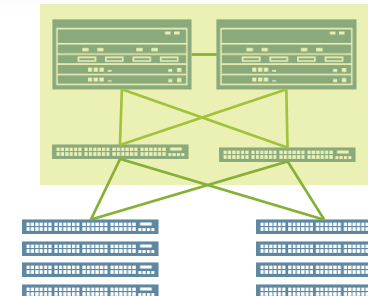
シンプル / 柔軟性

MC-LAG / ESI LAG



柔軟性 / 拡張性

EVPN-VXLAN / IP Fabric

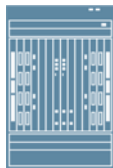


拡張性 / オープン

構成要素

コアスイッチ

EX9200



EX9250



アグリゲーションスイッチ



EX4600



EX4650 10/25GbE, 100GbE

アクセススイッチ



EX2300/EX3400



EX4100/EX4100-F



EX4300/EX4400



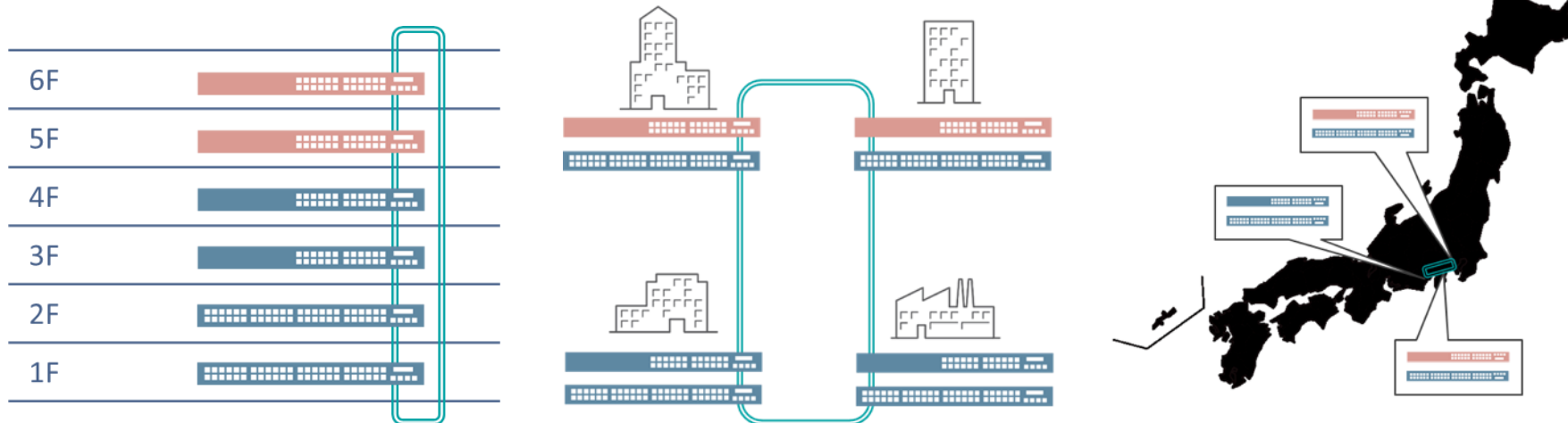
EX2300 2.5G



EX4300/EX4400
2.5/5/10GbE

Virtual Chassis (VC)

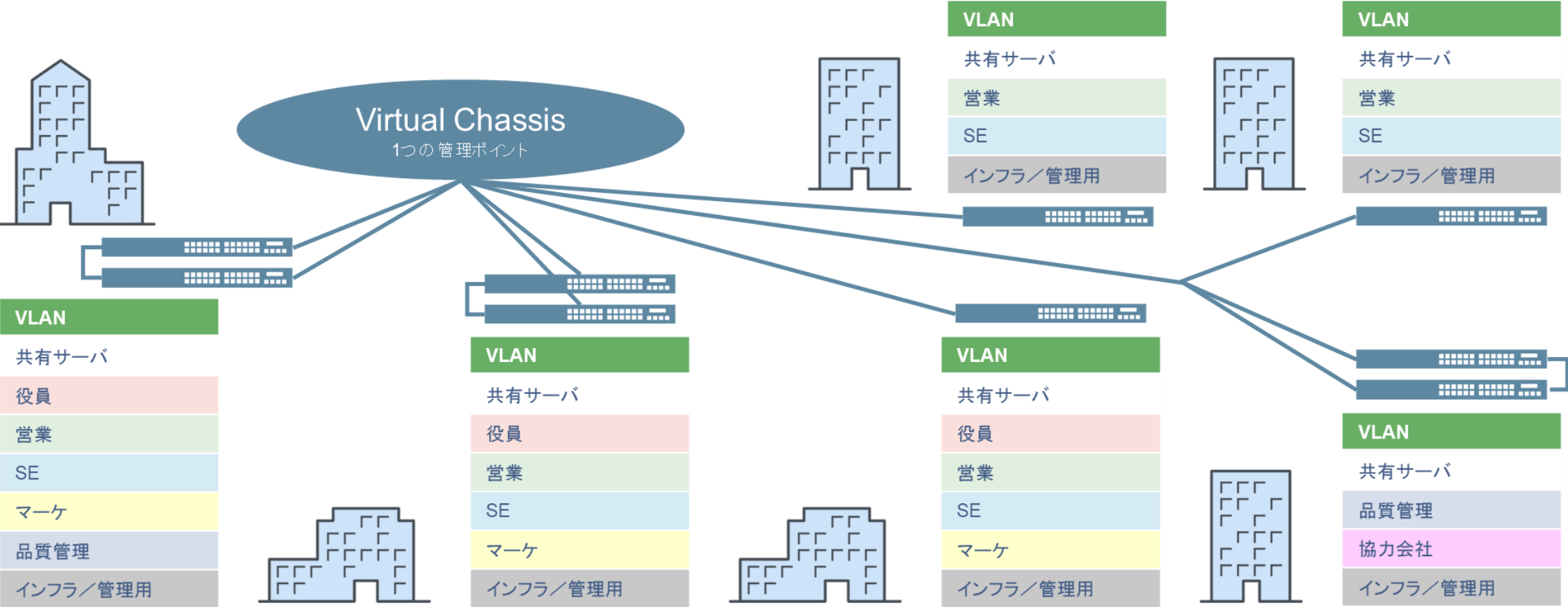
- 最大 10 台(※1)のスイッチを 1 台のスイッチにする仮想化テクノロジー
- 完全なループフリー構成になるため、スパニングツリー構成管理のわずらわしさから解放
- スwitch間接続インターフェースは通常の **SFP+ (10G)** や **QSFP+ (40G)** で、専用ケーブル不要の自由な物理構成
 - * 近距離接続用の安価な **DAC/AOC** を使用可能
- ダークファイバの有効活用 (1 芯接続可能な SFP+ をラインナップ)
- ハイアベイラビリティ機能で **L2/L3** のデータプレーンとコントロールプレーンを保護
 - **Non-Stop Bridging (NSB)**
 - **Non-Stop Routing (NSR)**
 - **Graceful Routing-Engine Switchover (GRES)**



※1; 製品により異なります

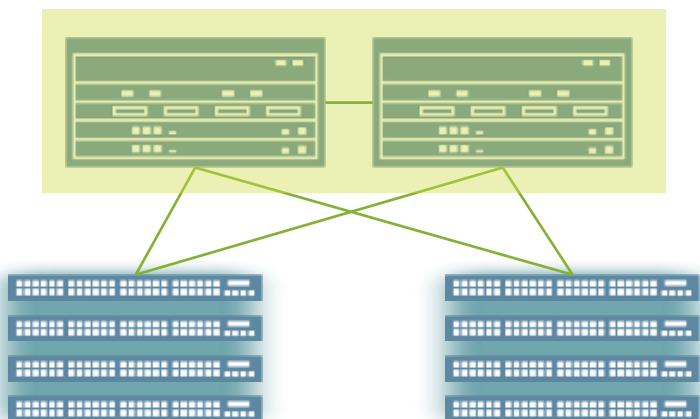
Virtual Chassis (VC)

建物や階はもちろん、広いエリアに散らばるスイッチを 1 台にする
スイッチ間の接続距離は最大 80km (10GBASE-ZR)



MC-LAG / ESI-LAG

MC-LAG or ESI-LAG



STP を必要としない

- 単一の仮想 L2/L3 インターフェースを提供
- HA/ active-active ロードバランスソリューション

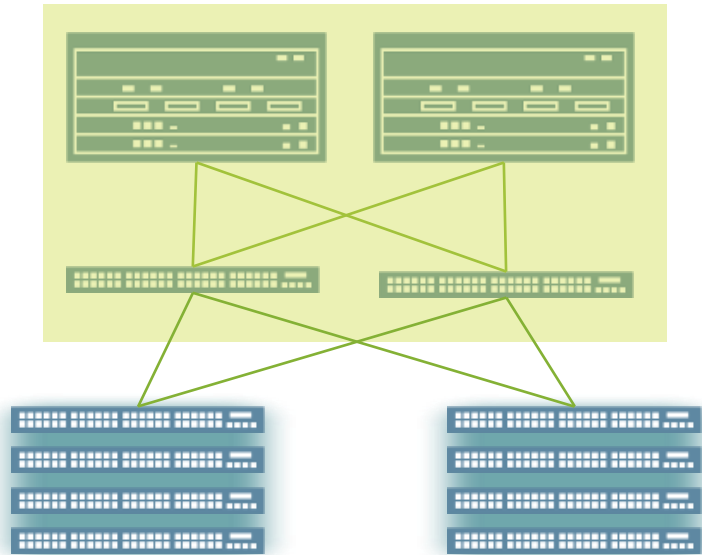
L3 インターフェースの MAC アドレスの同期

- VRRP を必要としない
- Essential for endpoint mobility

利点

- マルチホーミングの実現
- EVPN による延伸

IP Clos Fabric with EVPN VXLAN



課題

- スケーラブルな標準ベースのファブリック
- ファブリック全体で **L2** のモビリティが必要

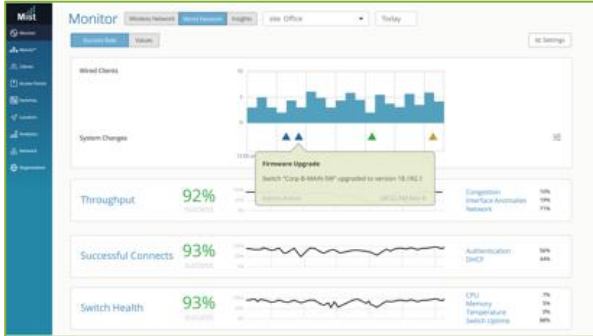
ソリューション

- **BGP** ベースの **IP fabric**
- **VXLAN** オーバレイによる **L2** ストレッチ
- コントロールプレーンは **EVPN**
- **Ansible** を用いた柔軟な自動化

利点

- **Active-active** マルチホーミング
- **Cloud** レベルの拡張性を備えた標準ベースの **IP** ファブリック
- リンクダウン時の高速コンバージェンス

管理と自動化



クラウドマネージメント

- [Juniper Mist Wired Assurance](#)
- ターゲット: SMB



オンプレミスマネージメント

- [Junos Space + Network Director / Security Director](#)
- [Juniper Connected Security \(former SDSN\)](#)
- ターゲット: ラージエンタープライズ



DIY

- EVPN-VXLAN
- Ansible プレイブック等を活用した自動化

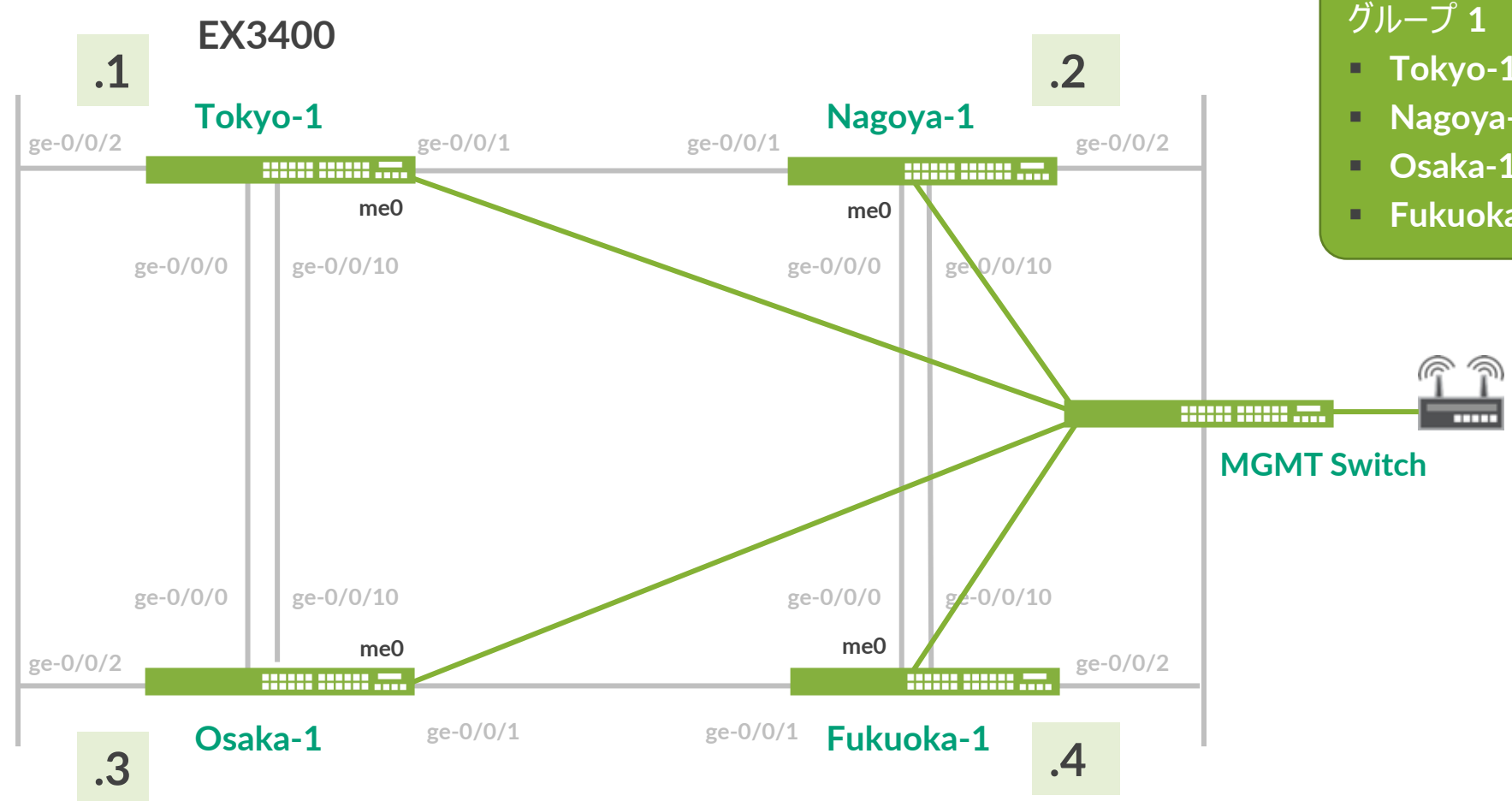


LAB.1

Junos の基本的な操作・設定

Ethernet Switching “EX/QFX” Course

Topology (Lab.1) – グループ 1



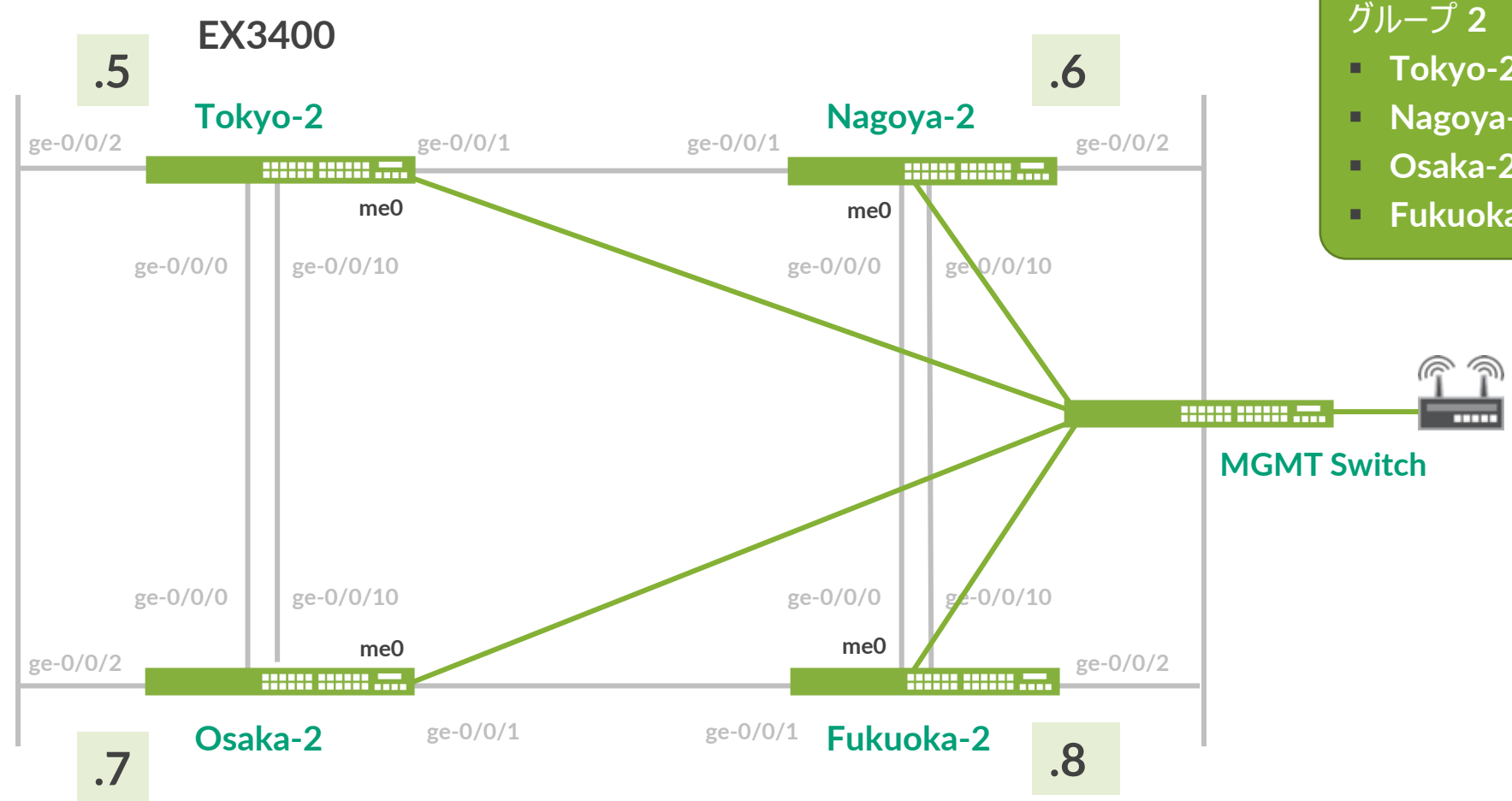
管理用 IP:
(me0) 192.168.1.x/24

グループ 1

- Tokyo-1: .1
- Nagoya-1: .2
- Osaka-1: .3
- Fukuoka-1: .4

Ethernet Switching “EX/QFX” Course

Topology (Lab.1) – グループ 2



管理用 IP:
(me0) 192.168.1.x/24

グループ 2

- Tokyo-2: .5
- Nagoya-2: .6
- Osaka-2: .7
- Fukuoka-2: .8

EX へのログイン

- 初期設定状態の EX にアカウント “root” でログイン
- CLI コマンドで Junos の Operational モードを起動
 - root アカウントは serial console、または SSH 接続のみ使用可能
 - 今回は事前に IP アドレス、root パスワード、SSH サービスが設定済みの状態
 - Tera Term から SSHv2 接続で接続してください

接続詳細	
IP アドレス :	192.168.1.x
サービス :	SSH (Tera Term)
ユーザ名 :	root
パスワード :	Juniper

```
--- JUNOS 20.2R3-S2.5 built 2021-07-30 09:45:37 UTC
root% cli
root>
```

Operational モードの show コマンド実行

構成やバージョンなど基本情報を確認

- Active Configuration を表示

```
root> show configuration
```

- ハードウェア情報を表示

```
root> show chassis hardware
```

- ソフトウェアバージョンの表示

```
root> show version
```

- インタフェースのステータス一覧の表示

```
root> show interface terse
```

- ルーティングテーブル表示

```
root> show route
```

- MAC アドレステーブル表示

```
root> show ethernet-switching table
```

- サポートを受ける際に必要な機器情報（RSI）を一括取得

```
root> request support information
```

※出力が一画面に入らない場合、| no-more オプションを追加すると最後まで一気に表示可能

root アカウントのパスワード設定（設定済）

- **Configuration** モードに入り、設定変更の準備を実施
- 下記の手順で **root** アカウントにパスワードを設定
 - root password : **Juniper**

```
root> configure
root# set system root-authentication plain-text-password
New password:
Retype new password:  (改行後パスワード入力 x2回)

[edit]
root# commit
```

※ **root** パスワード設定は必須です（設定が存在しないと **commit** がエラーとなる）

管理インタフェース（me0）へのアドレス付与（設定済）

管理インタフェース（me0）に対して管理用アドレスを付与

- アドレス **192.168.1.xx/24**（xx = **Topology** で指定された第 4 オクテット）を付与

```
# set interface me0 unit 0 family inet address 192.168.1.xx/24
```

EX3400 rear view



show interfaces コマンドで、設定した me0 インタフェースの Config を確認

```
{master:0}[edit]  
root# show interfaces
```

新規アカウント作成

管理用アカウント “lab” を以下の設定で作成

Username	Password	Class
lab	lab123	super-user

commit 完了後、一度 root ユーザのセッションをログアウト

```
root# set system login user lab class super-user
root# set system login user lab authentication plain-text-password
New password:
Retype new password:
[edit]
root# commit and-quit
root> exit
root@% exit
```

← (改行後パスワード入力 x2回)

SSH で、作成したアカウントを使って正常にログインできることを確認

```
--- JUNOS 20.2R3-S2.5 built 2021-07-30 09:45:37 UTC
lab>
```

サービスの起動とホスト名の設定

サービスの起動

- デフォルトでは各種サービスが起動していないため、追加で設定（SSH のみ事前に設定済み）
- **telnet**、**ftp**、**http** で機器にアクセスできるように設定

```
lab# set system services telnet
lab# set system services ftp
lab# set system services web-management http
```

ホスト名の作成（設定済み）

- **Topology** を参照して、各自がログインしている機器のホスト名を設定

```
lab# set system host-name EX-x
```

変更した Config の差分を確認

- **Active Config** と比較して、設定が正しく追加されたことを確認し **commit** を実行

```
lab# show | compare
lab# commit
```

サービス起動の確認

FTP によるアクセス

- **Windows** からコマンドプロンプトを立ち上げ **FTP** でアクセスできることを確認
 - **ftp 192.168.1.xx**
 - **root** を使用してログイン
 - **ls** コマンドでユーザディレクトリを表示できることを確認
- 表示されない場合、**Windows Firewall** で **FTP** 許可が必要

ブラウザから Web GUI (J-Web) へのアクセス

- ブラウザからアクセスし、**J-Web** の画面が表示されることを確認
- <http://192.168.1.xx/>
- **root**、または作成したユーザ (**lab**) を使用してログイン

J-Web GUI

シンプルで直観的な操作が可能なウェブ管理インタフェース



J-Web
NoName

Dashboard Configure Monitor Maintain Troubleshoot Commit Help

Configure

- Interfaces
- Ports
- Link Aggregation
- Switching

Port Configuration

List ports for FPC: All

Port	Link Status	Type	Port Role	VLAN (VLAN ID)	Description
ge-0/0/0	Down	Gigabit Ethernet	None	default (1)	
ge-0/0/1	Down	Gigabit Ethernet	None	default (1)	
ge-0/0/2	Down	Gigabit Ethernet	None	default (1)	
ge-0/0/3	Down	Gigabit Ethernet	None	default (1)	
0/4	Down	Gigabit Ethernet	None	default (1)	
0/5	Down	Gigabit Ethernet	None	default (1)	
0/6	Down	Gigabit Ethernet	None	default (1)	
0/7	Down	Gigabit Ethernet	None	default (1)	
0/8	Down	Gigabit Ethernet	None	default (1)	
0/9	Down	Gigabit Ethernet	None	default (1)	
0/10	Down	Gigabit Ethernet	None	default (1)	
0/11	Down	Gigabit Ethernet	None	default (1)	

Configuration of port: ge-0/0/0

Property	Value
Administrative Status	Up
Physical Interface	ge-0/0/0.0
Mode	access
VLAN (VLAN ID)	None
Address/Subnet Mask	-
Address/Subnet Mask (bytes)	1514
Speed	Auto-Negotiation
Duplex	Automatic

Copyright © 2019, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.

Configuration の確認

ここまでで設定した **Configuration** 全体を確認

- ① **Operational** モードから確認
稼働中の **Active Config** を表示

```
lab@Tokyo-1> show configuration  
lab@Tokyo-1> show configuration | display set
```

} 同じ **Config** を異なる形式で表示

- ② **Configuration** モードから確認
編集中の **Candidate Config** を表示
commit 後に設定変更をしていなければ、**Active Config** と同じ内容が表示される

```
lab@Tokyo-1> configure  
Entering configuration mode
```

```
[edit]  
lab@Tokyo-1# show  
lab@Tokyo-1# show | display set
```

} 同じ **Config** を異なる形式で表示

Operational モードのコマンドを表示

Configuration モードから、Operational モードのコマンドを実行

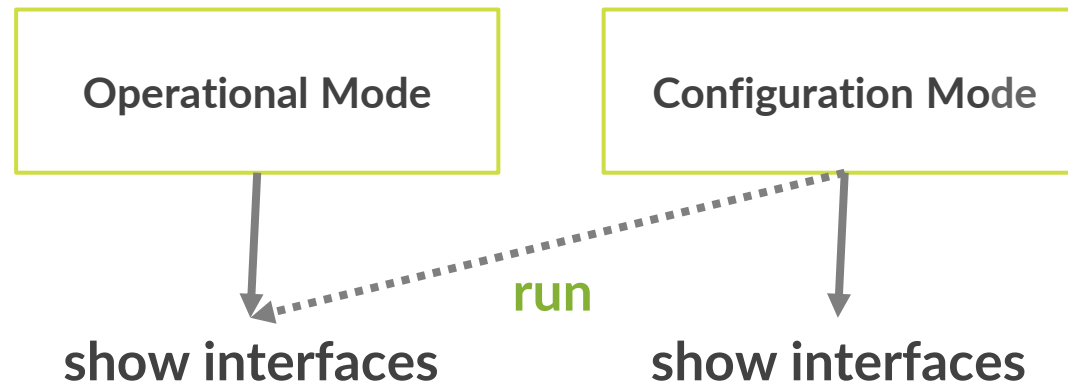
① Configuration モードにアクセス

```
lab@Tokyo-1> configure
```

② show interfaces コマンドを実行

以下の 2 つのコマンドを実行し、表示される内容を確認

```
lab@Tokyo-1# show interfaces  
lab@Tokyo-1# run show interfaces
```



commit confirmed

誤った設定をしてしまった場合でも設定が自動で元に戻ることを確認

① コマンドプロンプトから **ping 192.168.1.xx -t** を継続して実行

② 管理インタフェースの設定を削除

me0 の設定を削除 ※ **commit** はまだしないこと

```
lab@Tokyo-1# delete interfaces me0
lab@Tokyo-1# show | compare
```

③ **commit confirmed**

commit confirmed オプションを使って、1 分後に設定が戻るように **commit**

commit 完了メッセージが表示された後、アクセス不能になり Tera Term が切断動作になる

```
lab@Tokyo-1# commit confirmed 1
```

④ **ping** 応答が返ってきたら再度ログインし、設定が戻っていることを確認

削除したインターフェースの設定がもとに戻っていることを確認

```
lab@Tokyo-1> show configuration interfaces me0
```

Configuration をファイルに保存

- 次の Lab を始める前に、**save** コマンドで **Configuration File** を保存
- **file list** コマンドで正常に **save** できたことを確認

```
lab@Tokyo-1# save lab1-end_YYMMDD
Wrote 213 lines of configuration to 'lab1-end_YYMMDD'

[edit]
lab@Tokyo-1# exit
Exiting configuration mode

lab@Tokyo-1> file list

/var/home/lab/:
.ssh/
lab1-end_YYMMDD
```



LAB.2

Interface の設定

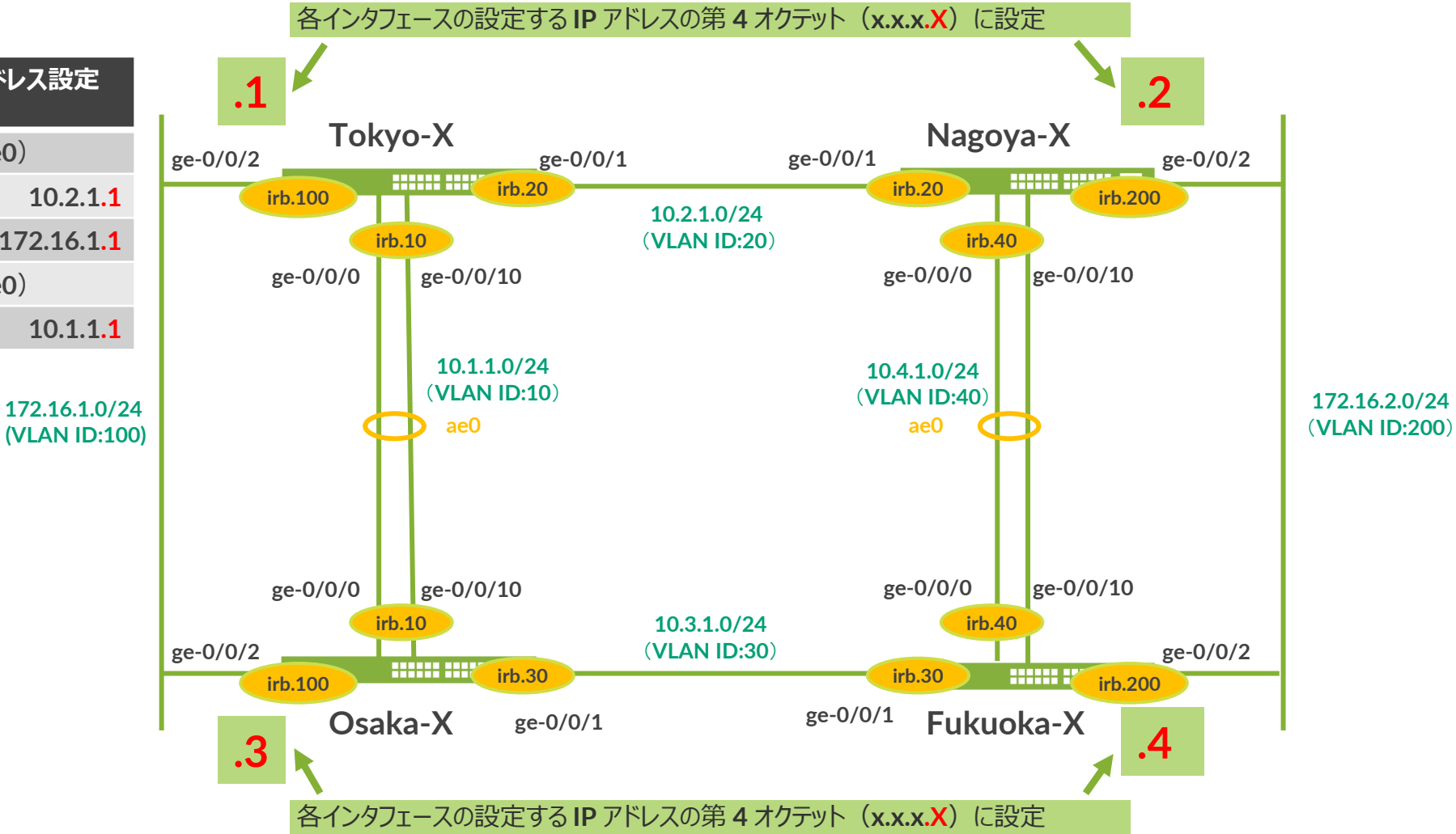
Link Aggregation / VLAN / IRB

Ethernet Switching “EX/QFX” Course

Topology (Lab.2 : インタフェースの設定)

例 : Tokyo-X の場合の IP アドレス設定

ge-0/0/0	(ae0)
ge-0/0/1 (irb.20)	10.2.1.1
ge-0/0/2 (irb.100)	172.16.1.1
ge-0/0/10	(ae0)
ae0 (irb.10)	10.1.1.1



LAG (Link Aggregation Group) の作成 ①

LAG (Link Aggregation Group) を作成

- LAG の設定準備

```
# set chassis aggregated-devices ethernet device-count 1
```

- LAG の作成 (ae0)

```
# set interfaces ae0 unit 0 family ethernet-switching
```

- LAG に参加させるメンバー IF の default protocol-family (ethernet-switching) を削除

```
# delete interfaces ge-0/0/0 unit 0  
# delete interfaces ge-0/0/10 unit 0
```

- LAG のメンバー IF へ ae0 追加

```
# set interfaces ge-0/0/0 ether-options 802.3ad ae0  
# set interfaces ge-0/0/10 ether-options 802.3ad ae0
```

※ LAG の場合、論理インタフェースプロパティは ae インタフェースに対して設定
メンバー IF には設定しない (メンバー IF は論理 IF を持たない)

LAG (Link Aggregation Group) の作成 ②

LAG に LACP を設定

- LACP オプションの追加し、commit

```
# set interfaces ae0 aggregated-ether-options lacp active periodic fast
```

LAG の正常性を確認

- LAG の状態を確認

```
> show interfaces ae0  
> show interface terse
```

- ae0 が Admin up、Link up のステータスであること
- LACP の状態を確認

```
> show lacp interfaces ae0
```

- LACP protocol が以下の状態になっていること
 - Receive State: Current
 - Mux State: Collecting distributing

VLAN を作成し、アクセスポートを設定

VLAN を作成し、インタフェースへの適用を行います

- VLAN 作成

- Topology を参照し、機器が所属する VLAN を作成する

```
lab@Tokyo# set vlans vlan10 vlan-id 10
lab@Tokyo# set vlans vlan20 vlan-id 20
lab@Tokyo# set vlans vlan100 vlan-id 100
```

- インタフェースを access port に設定し、VLAN に参加させる

```
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching interface-mode access
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching vlan members vlan10

lab@Tokyo# set interface ge-0/0/1 unit 0 family ethernet-switching interface-mode access
lab@Tokyo# set interface ge-0/0/1 unit 0 family ethernet-switching vlan members vlan20

lab@Tokyo# set interface ge-0/0/2 unit 0 family ethernet-switching interface-mode access
lab@Tokyo# set interface ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```


VLAN & Interface: Sample Configuration

- Tokyo

```
set vlans vlan10 vlan-id 10
set vlans vlan20 vlan-id 20
set vlans vlan100 vlan-id 100

set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan20
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```

- Nagoya

```
set vlans vlan20 vlan-id 20
set vlans vlan40 vlan-id 40
set vlans vlan200 vlan-id 200

set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan20
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan40
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan200
```

VLAN & Interface: Sample Configuration

- Osaka

```
set vlans vlan10 vlan-id 10
set vlans vlan30 vlan-id 30
set vlans vlan100 vlan-id 100

set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan30
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```

- Fukuoka

```
set vlans vlan30 vlan-id 30
set vlans vlan40 vlan-id 40
set vlans vlan200 vlan-id 200

set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan30
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan40
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan200
```

VLAN に IRB インタフェースを追加

VLAN に IRB （Integrated Routing and Bridging = L3 vlan interface）を設定

VLAN インタフェースにアドレスを追加

- **Topology** を参照し、該当する **VLAN** インタフェースを作成し、アドレスを設定

```
lab@Tokyo# set interfaces irb unit 10 family inet address 10.1.1.x/24
lab@Tokyo# set interfaces irb unit 20 family inet address 10.2.1.x/24
lab@Tokyo# set interfaces irb unit 100 family inet address 172.16.1.x/24
```

- **VLAN** に対して **L3** インタフェースをひもづける

```
lab@Tokyo# set vlans vlan10 l3-interface irb.10
lab@Tokyo# set vlans vlan20 l3-interface irb.20
lab@Tokyo# set vlans vlan100 l3-interface irb.100
```

RSTP を無効化（delete）

- デフォルトで動作している **RSTP** は必要なくなったため、設定を削除

```
lab@Tokyo# delete protocols rstp
```

IRB: Sample Configuration

- Tokyo

```
set interfaces irb unit 10 family inet address 10.1.1.1/24
set interfaces irb unit 20 family inet address 10.2.1.1/24
set interfaces irb unit 100 family inet address 172.16.1.1/24

set vlans vlan10 l3-interface irb.10
set vlans vlan20 l3-interface irb.20
set vlans vlan100 l3-interface irb.100

delete protocols rstp
```

- Nagoya

```
set interfaces irb unit 20 family inet address 10.2.1.2/24
set interfaces irb unit 40 family inet address 10.4.1.2/24
set interfaces irb unit 200 family inet address 172.16.2.2/24

set vlans vlan20 l3-interface irb.20
set vlans vlan40 l3-interface irb.40
set vlans vlan200 l3-interface irb.200

delete protocols rstp
```

IRB: Sample Configuration

- **Osaka**

```
set interfaces irb unit 10 family inet address 10.1.1.3/24
set interfaces irb unit 30 family inet address 10.3.1.3/24
set interfaces irb unit 100 family inet address 172.16.1.3/24

set vlans vlan10 l3-interface irb.10
set vlans vlan30 l3-interface irb.30
set vlans vlan100 l3-interface irb.100

delete protocols rstp
```

- **Fukuoka**

```
set interfaces irb unit 30 family inet address 10.3.1.4/24
set interfaces irb unit 40 family inet address 10.4.1.4/24
set interfaces irb unit 200 family inet address 172.16.2.4/24

set vlans vlan30 l3-interface irb.30
set vlans vlan40 l3-interface irb.40
set vlans vlan200 l3-interface irb.200

delete protocols rstp
```

インタフェース / VLAN / LACP の動作確認

インタフェース、VLAN、LACP の確認コマンドで正常性を確認

```
> show interfaces (terse)
> show ethernet-switching interfaces
> show vlans (detail)
> show lacp interfaces
```

隣接機器に対して **ping** を実施し、応答があることを確認

- **ping** [隣接機器の IRB IP アドレス]
- **Ctrl+C** で停止

※参考：VLAN を作成し、トランクポートを設定する場合

VLAN を作成し、インタフェースへの適用を行います

- VLAN 作成

```
lab@Tokyo# set vlans vlan10 vlan-id 10
lab@Tokyo# set vlans vlan20 vlan-id 20
lab@Tokyo# set vlans vlan100 vlan-id 100
```

- インタフェースをトランクポートに設定し、複数の VLAN を参加させる

```
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching interface-mode trunk
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching vlan members vlan10
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching vlan members vlan20
lab@Tokyo# set interface ae0 unit 0 family ethernet-switching vlan members vlan100
```

※本トレーニングコースの構成にてトランクポートの設定が行えないため、
参考までに設定方法を記載しています（**トレーニング中は、実機に投入しないでください**）

※参考： Legacy Layer2 Switching モデルの場合

Legacy (旧) L2 Switching モデル (SRX100 ～ SRX650、EX2200 ～ EX4550 など) では関連の設定コマンドは以下になる

```
set vlans vlan10 vlan-id 10
set vlans vlan20 vlan-id 20
set vlans vlan100 vlan-id 100

set interfaces ae0 unit 0 family ethernet-switching port-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan10
set interfaces ge-0/0/1 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan20
set interfaces ge-0/0/2 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100

set vlans vlan10 l3-interface vlan.10
set vlans vlan20 l3-interface vlan.20
set vlans vlan100 l3-interface vlan.100

set interfaces vlan unit 10 family inet address 10.1.1.1/24
set interfaces vlan unit 20 family inet address 10.2.1.1/24
set interfaces vlan unit 100 family inet address 172.16.1.1/24

delete protocols rstp
```



LAB.3

Routing の設定

OSPF / Redistribute Static

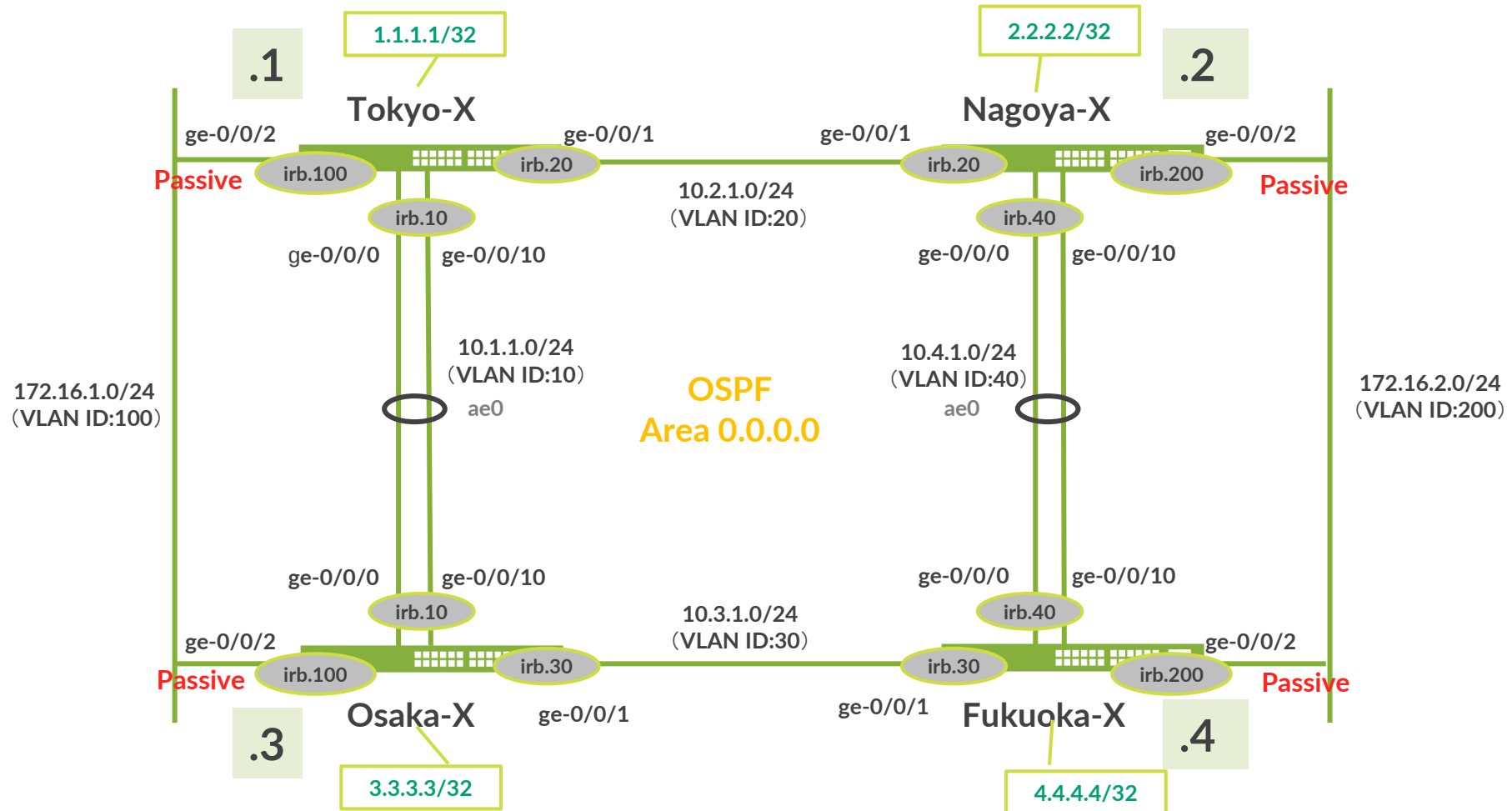
Ethernet Switching “EX/QFX” Course

Topology (Lab.3 : ルーティングの設定)

Loopback address
(全グループ共通)

Tokyo : 1.1.1.1/32 Nagoya : 2.2.2.2/32

Osaka : 3.3.3.3/32 Fukuoka : 4.4.4.4/32



OSPF でのルーティング

OSPF の設定を行います

- Loopback アドレス (lo0) を設定する

```
lab@Tokyo# set interfaces lo0 unit 0 family inet address 1.1.1.1/32
```

- Router ID を設定する

```
lab@Tokyo# set routing-options router-id 1.1.1.1
```

- OSPF に参加させたいインタフェースを追加する

```
lab@Tokyo# set protocols ospf area 0 interface lo0.0  
lab@Tokyo# set protocols ospf area 0 interface irb.10  
lab@Tokyo# set protocols ospf area 0 interface irb.20
```

- Passive インタフェースを設定する

```
lab@Tokyo# set protocols ospf area 0 interface irb.100 passive
```

OSPF: Sample Configuration

- Tokyo

```
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set routing-options router-id 1.1.1.1
set protocols ospf area 0 interface lo0.0
set protocols ospf area 0 interface irb.10
set protocols ospf area 0 interface irb.20
set protocols ospf area 0 interface irb.100 passive
```

- Nagoya

```
set interfaces lo0 unit 0 family inet address 2.2.2.2/32
set routing-options router-id 2.2.2.2
set protocols ospf area 0 interface lo0.0
set protocols ospf area 0 interface irb.20
set protocols ospf area 0 interface irb.40
set protocols ospf area 0 interface irb.200 passive
```

- Osaka

```
set interfaces lo0 unit 0 family inet address 3.3.3.3/32
set routing-options router-id 3.3.3.3
set protocols ospf area 0 interface lo0.0
set protocols ospf area 0 interface irb.10
set protocols ospf area 0 interface irb.30
set protocols ospf area 0 interface irb.100 passive
```

- Fukuoka

```
set interfaces lo0 unit 0 family inet address 4.4.4.4/32
set routing-options router-id 4.4.4.4
set protocols ospf area 0 interface lo0.0
set protocols ospf area 0 interface irb.30
set protocols ospf area 0 interface irb.40
set protocols ospf area 0 interface irb.200 passive
```

OSPF の動作確認 ①

OSPF のネイバーのステータスを確認

```
lab@Tokyo> show ospf neighbor
```

DR / BDR の確認

```
lab@Tokyo> show ospf interface
```

OSPF 経由で学習した経路を表示

```
lab@Tokyo> show route
```

Loopback アドレスを送信元として ping 、 traceroute を実行し、全体に疎通できることを確認

```
lab@Tokyo> ping < 宛先 address > source < 自機の Lo0 address >  
lab@Tokyo> traceroute < 宛先 address >
```

OSPF の動作確認 ②

OSPF 経由で学習した経路のみを表示

```
lab@Tokyo> show route protocol ospf
```

OSPF データベース（ AREA ごと）

```
lab@Tokyo> show ospf database area 0
```

ルータがアドバタイズしている LSA を表示

```
lab@Tokyo> show ospf database router advertising-router < 対向 router-id > detail
```


OSPF でのルーティング

Static Route の OSPF への Redistribute

- Dummy の Static Route を設定

```
lab@Tokyo# set routing-options static route 9.9.9.X discard
```

- Static Route を Export する Policy を作成

```
lab@Tokyo# set policy-options policy-statement EXPORT-OSPF from protocol static  
lab@Tokyo# set policy-options policy-statement EXPORT-OSPF then accept
```

- OSPF に Export Policy を適用

```
lab@Tokyo# set protocols ospf export EXPORT-OSPF
```



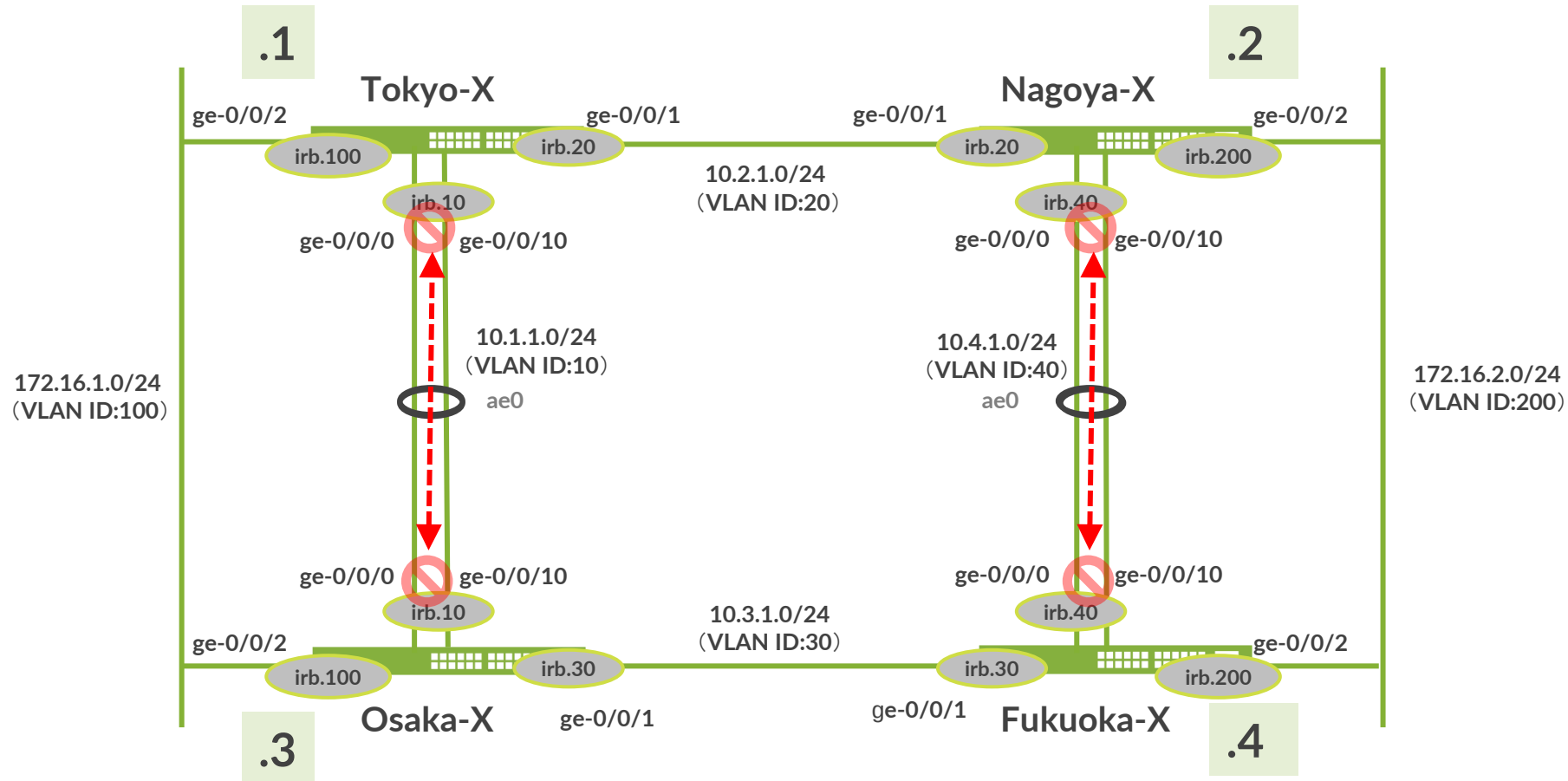
LAB.4

Firewall Filter (ACL) の設定

Ethernet Switching “EX/QFX” Course

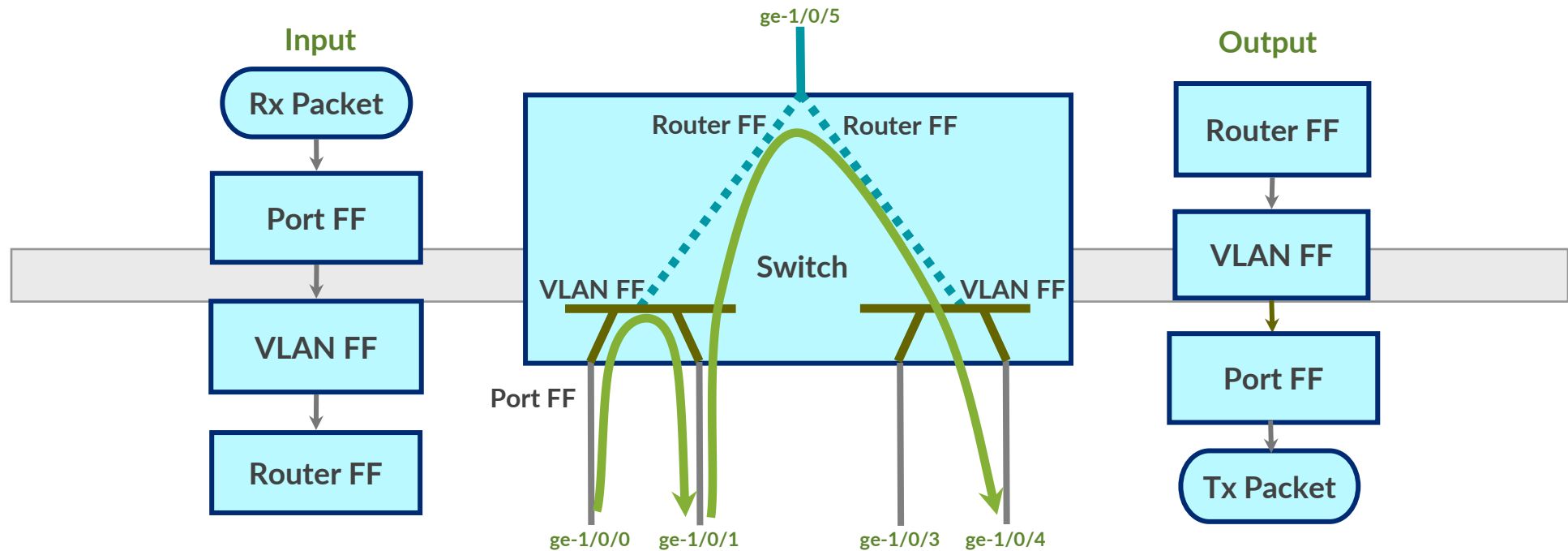
Topology (Lab.4 : アクセスリストの設定)

- ① 各 EX で図のように EX から EX へ、Telnet アクセスを禁止してください
- ② me0 に Filtering をかけて、FTP アクセスを禁止してください



Firewall Filter チェック順序

- Firewall Filter のチェック順序
 - Port → VLAN → Router の順序にて FF を実行 （ Egress は逆の手順にて実行）
- Router の FF は、同一 VLAN 内の Switch パケットに適用できない



Firewall Filter 設定

Port/VLAN-based FF

```
firewall {  
    family ethernet-switching {  
        filter <filter-name> {  
            term <term-name> {  
                from {  
                    <match conditions>;  
                }  
                then <actions defined>;  
            }  
            term implicit-rule {  
                then discard;  
            }  
        }  
    }  
}
```

Router-based FF

```
firewall {  
    family inet {  
        filter <filter-name> {  
            term <term-name> {  
                from {  
                    <match conditions>;  
                }  
                then <actions defined>;  
            }  
            term implicit-rule {  
                then discard;  
            }  
        }  
    }  
}
```

- パケットは、**term** の上位からルックアップされる
- マッチした **term** の **action** を実行してぬける
- 最後に暗黙の **deny** (**implicit-rule**) が隠れている

① Firewall Filter を使用して Telnet を制御

Firewall Filter を設定

```
set firewall family ethernet-switching filter deny_telnet term t10 from ip-protocol tcp
set firewall family ethernet-switching filter deny_telnet term t10 from destination-port telnet
set firewall family ethernet-switching filter deny_telnet term t10 then discard
set firewall family ethernet-switching filter deny_telnet term t10 then count telnet_count
set firewall family ethernet-switching filter deny_telnet term t20 then accept
```

Filter を ae0 インタフェースへ適用

```
set interfaces ae0 unit 0 family ethernet-switching filter input deny_telnet
```

Filter が有効なことを確認するため、telnet で隣接機器にアクセス

```
lab@Tokyo> telnet 10.1.1.3
Trying 10.1.1.3...
[Ctrl+C で停止]
```

telnet 以外の通信（ping、OSPF）は依然可能なことを確認

Filter で discard された telnet のカウンタを確認

```
lab@Tokyo> show firewall
```

② Firewall Filter を使用して管理インタフェースを制御

Firewall Filter を設定

```
set firewall family inet filter deny_ftp term t10 from protocol tcp
set firewall family inet filter deny_ftp term t10 from destination-port ftp
set firewall family inet filter deny_ftp term t10 then discard
set firewall family inet filter deny_ftp term t10 then count ftp_count
set firewall family inet filter deny_ftp term t20 then accept
```

Filter をインタフェースへ適用する

```
set interfaces lo0 unit 0 family inet filter input deny_ftp
set interfaces me0 unit 0 family inet filter input deny_ftp
```

コマンドプロンプトから FTP で管理 IP アドレスにアクセスできなくなったことを確認

Filter で **discard** された **ftp** のカウンタを確認

```
lab@Tokyo> show firewall
```

※ EX、QFX シリーズ自身への通信を制御する場合、lo0 および me0（EX）、em0（QFX）へ FF を適用する必要がある

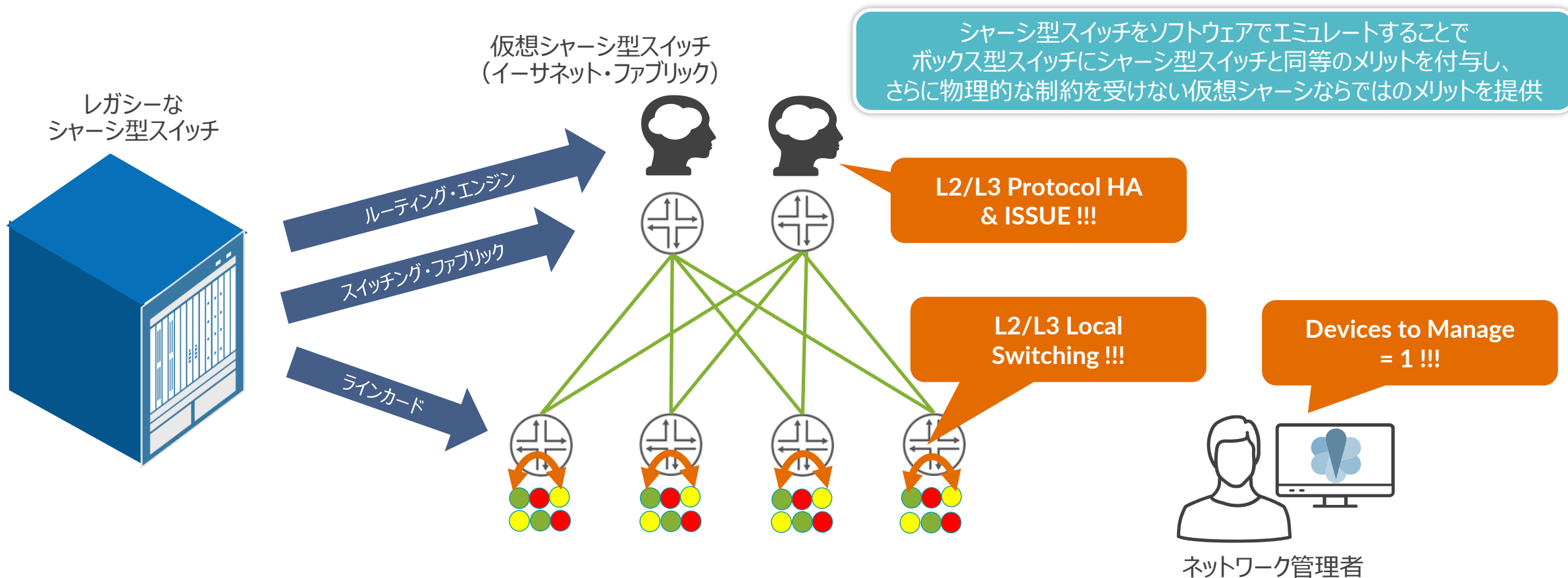
※ SRX、MX シリーズ自身への通信を制御する場合、lo0 のみに Firewall Filter を適用することで制御可能
（管理インタフェース fxp0 への適用は不要）



Virtual Chassis とは

ジュニパーのイーサネット・ファブリック

ジュニパーの解決策：
ソフトウェアの力で仮想的にシャーシ型スイッチをエミュレート



旧来のシャーシ型スイッチと Virtual Chassis 技術

シャーシ型スイッチのメリット

✓高信頼性ハードウェア

- 冗長ルーティングエンジン
- 冗長スイッチファブリック
- 冗長電源ユニット
- 冗長ファントレイ

✓管理の簡便性

- シングルイメージ
- 単一のコンフィグファイル
- 単一のマネージメントIPアドレス

✓パフォーマンスとスケーリング

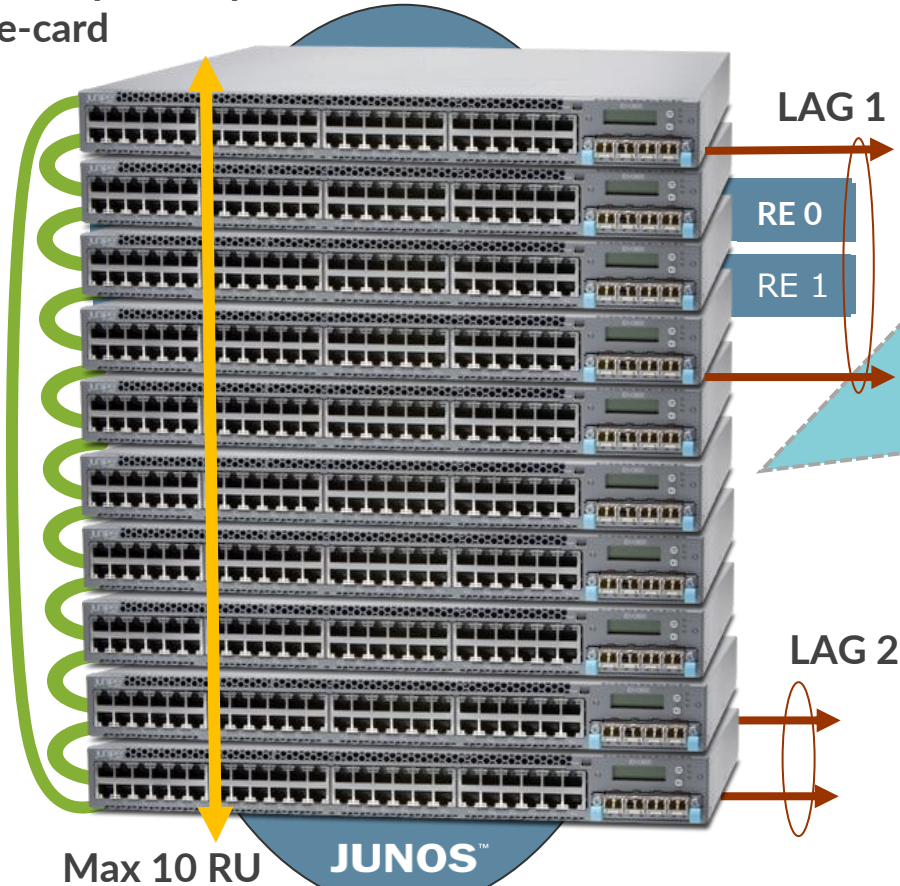
- ハイパフォーマンス
- 大容量のバックプレーン
- モジュラー型構成



✓Virtual Chassis による更なるメリット:

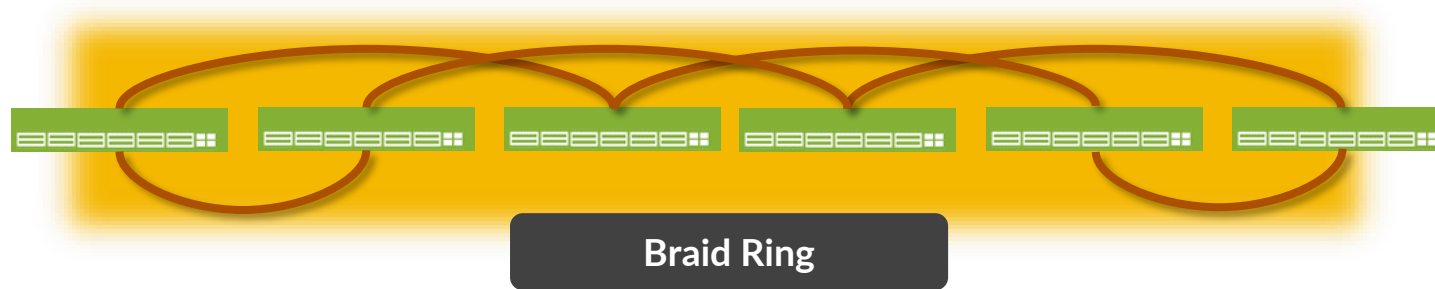
- 物理配置の柔軟性
- 低消費電力
- 最小構成からスタート可能
- 必要最低限のラックスペース確保

Max 480Gbps Backplane
Per line-card

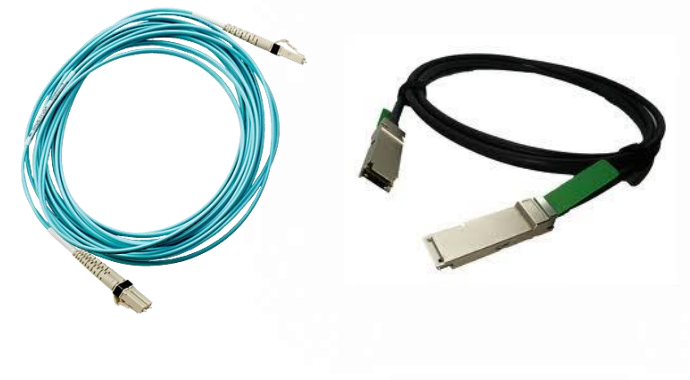
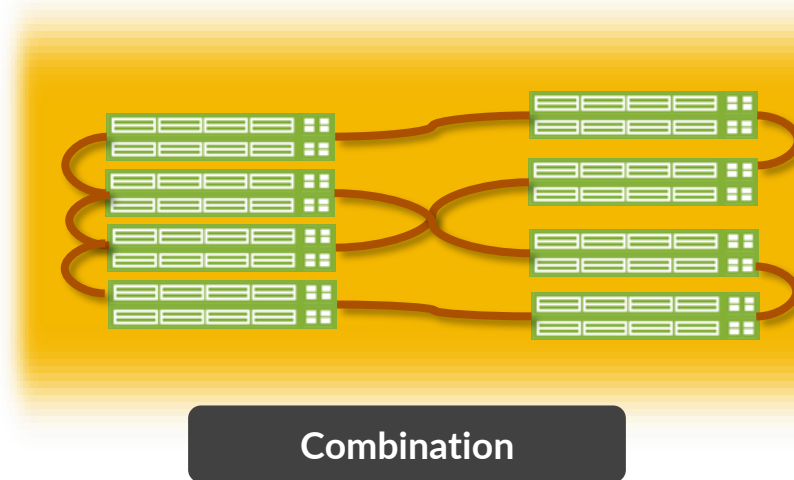
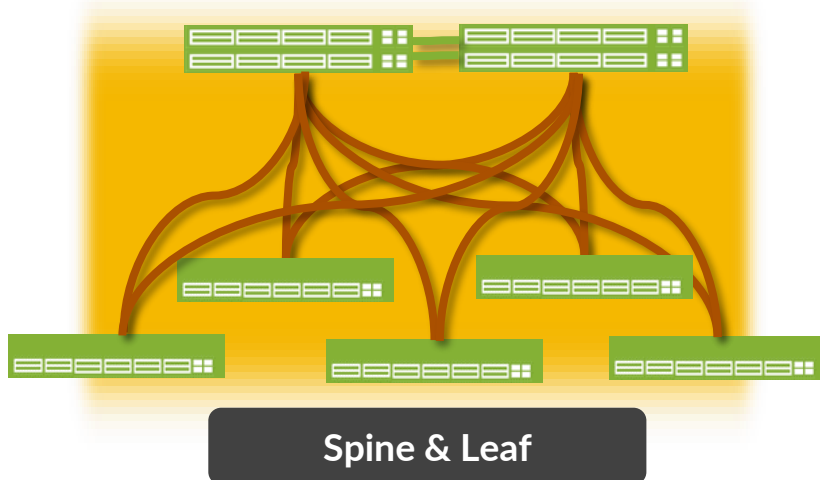


Virtual Chassis のトポロジー

最大 10 メンバーまでであれば、仮想バックプレーンによる接続を使用した自由なトポロジーで L2/L3 ファブリックを構成することが可能



※接続方法は、筐体間の距離に応じて
Copper (DAC/QSFP-DAC) か Fiber から選択



Virtual Chassis の仮想バックプレーン

前面・背面のファイバー・イーサネット・ポートを自由に
仮想バックプレーンに変換して **VC** を構成することが可能

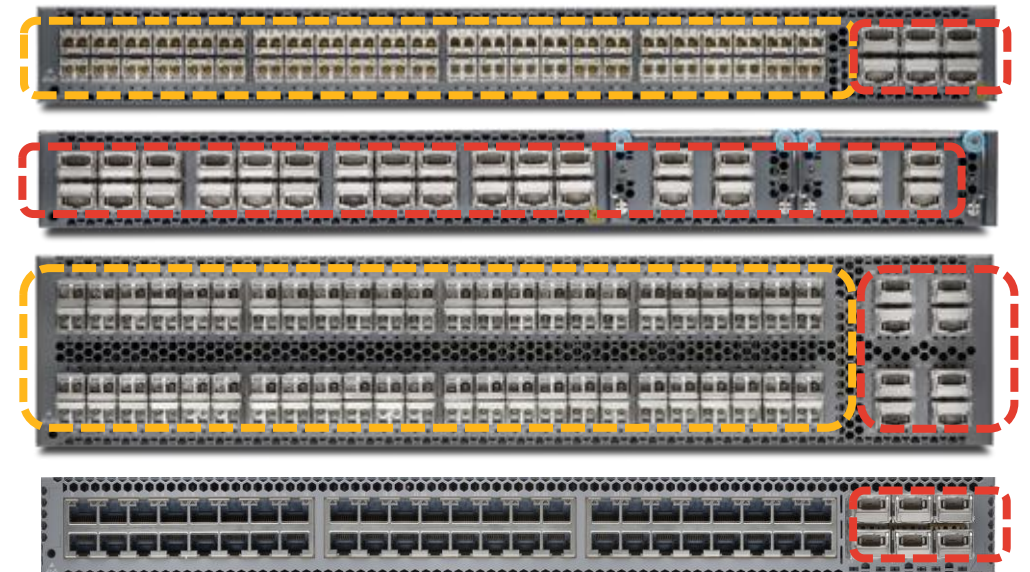
EX3400 シリーズ



EX4300 シリーズ



QFX5100 シリーズ



 10G*N Gbps 仮想バックプレーン (VC ポート)

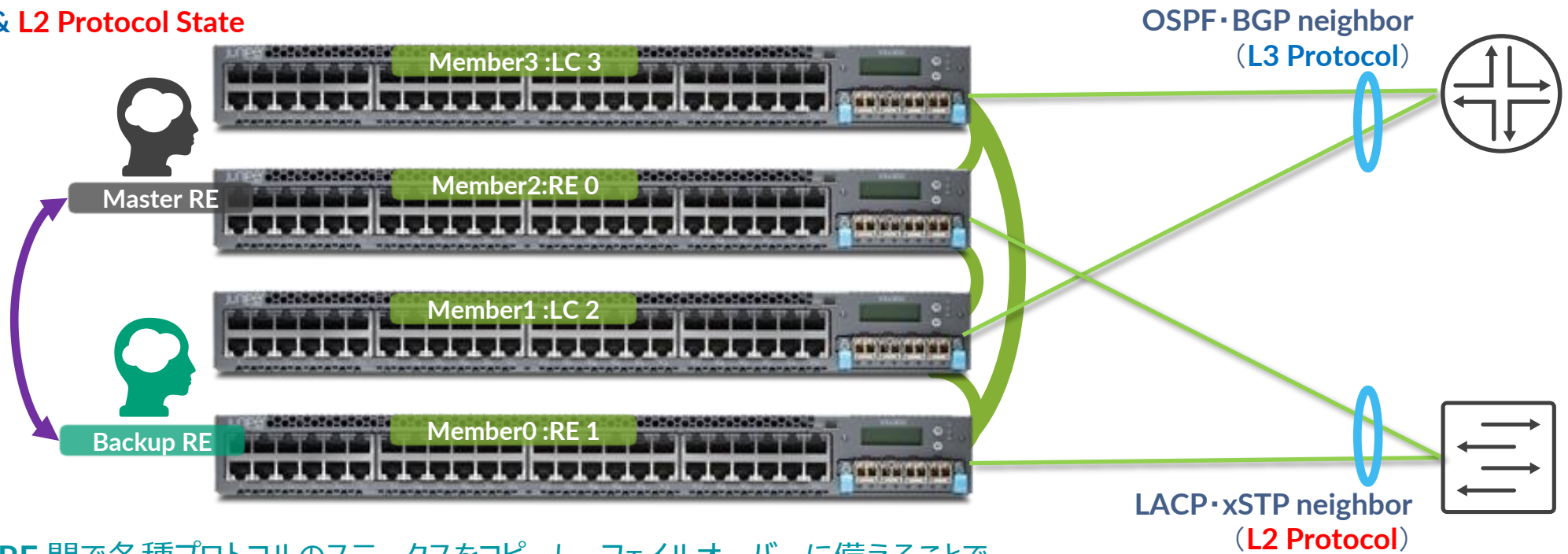
 40G*N Gbps 仮想バックプレーン (VC ポート)

Virtual Chassis の High Availability 機能

Master RE に障害が発生しても無停止でプロトコル継続運用が可能な
Non Stop Routing (NSR) および Non Stop Bridging (NSB)

Kernel、Fowarding Table、interface info

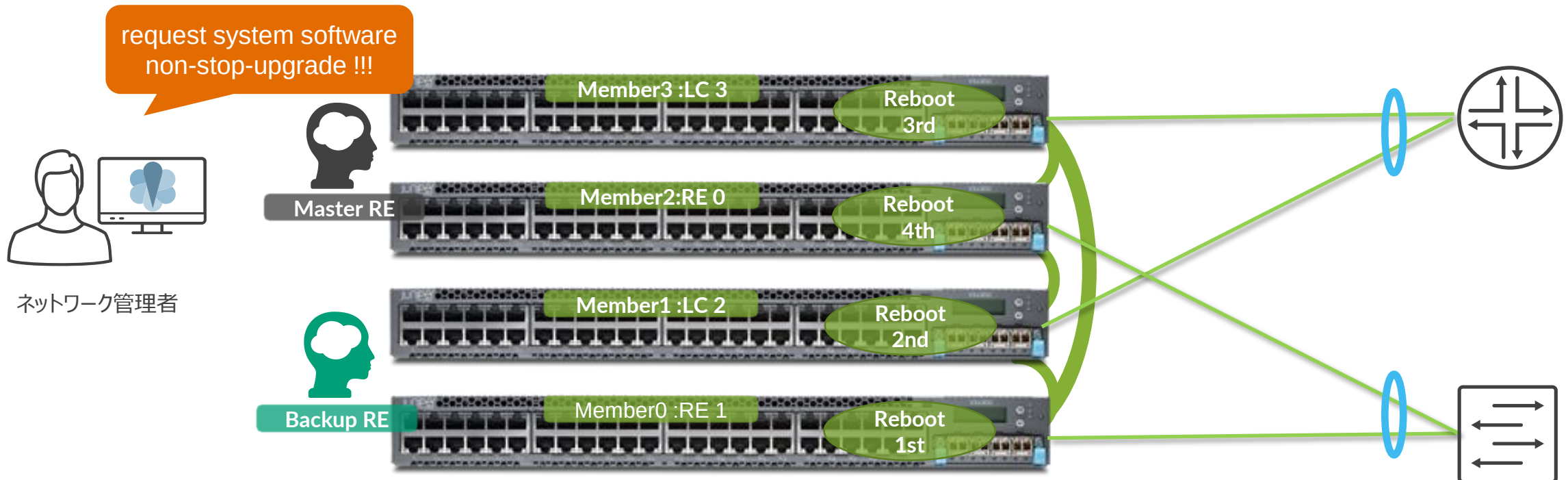
L3 Protocol State & L2 Protocol State



RE 間で各種プロトコルのステータスをコピーし、フェイルオーバーに備えることで
障害時における L2/L3 プロトコルへの影響を最小化

Virtual Chassis の OS バージョンアップ

Non Stop Software Upgrade (NSSU) により管理者は、コマンド一行でシステムダウンタイムを約 1 秒以内の想定※で OS のバージョンアップを実行

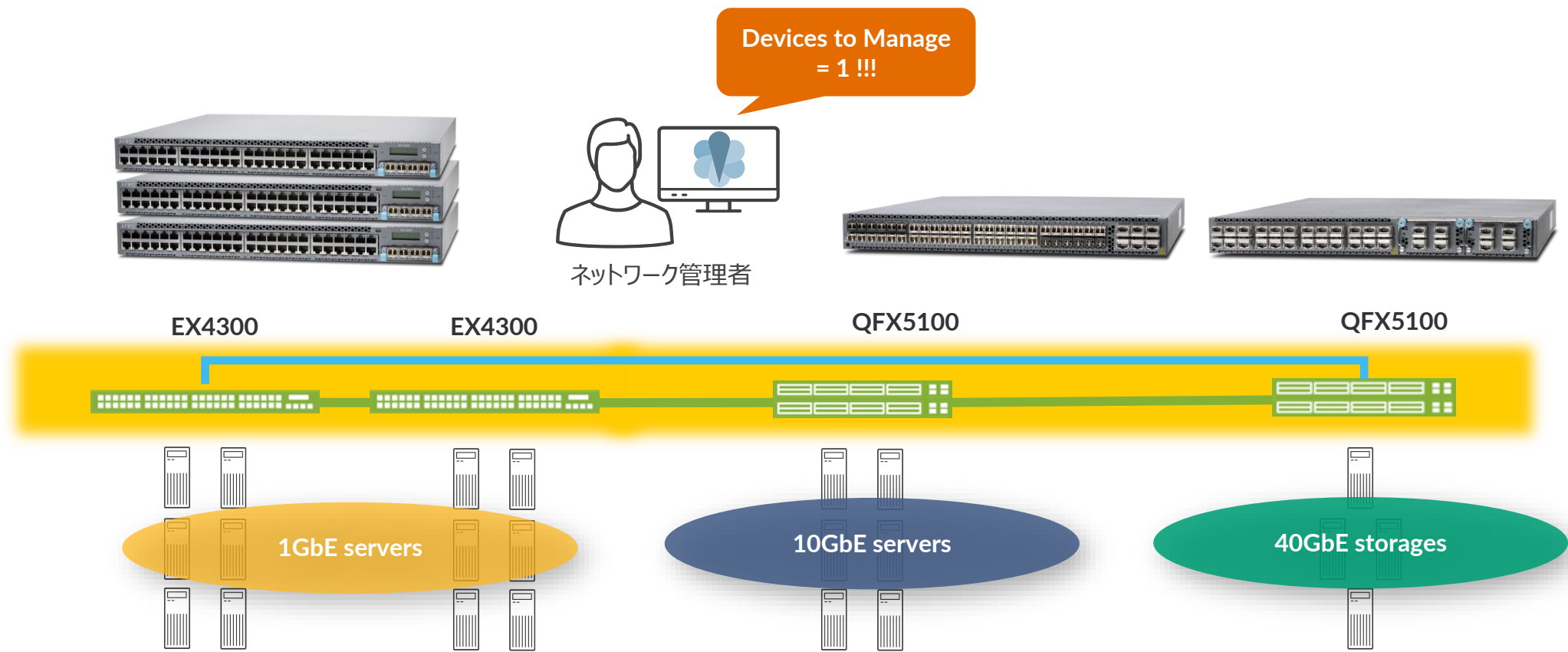


管理者による Upgrade コマンドの発呼後、各 RE、Linecard と順に再起動して新 OS を反映するため、ラインカード跨ぎのインタフェースの保護構成を取ることと NSR/NSB との併用で OS アップグレード時の影響を最小化することが可能

※すべての環境で 1 秒以内のダウンタイムを保証するものではありません 環境に応じた事前の検証を推奨しております

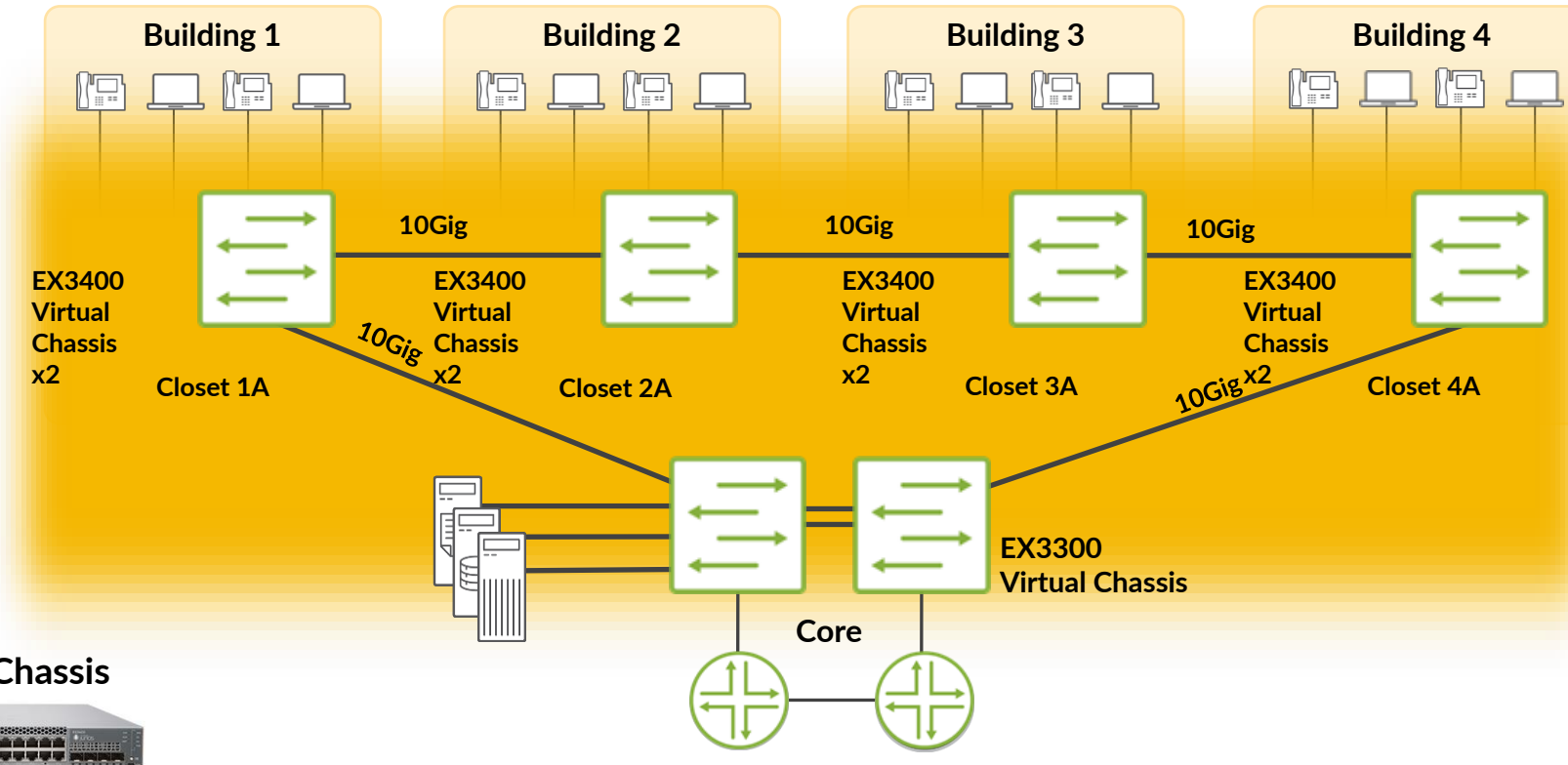
Virtual Chassis Mixed Mode

異なるメディアスピードのラインカードを **Virtual Chassis** 内で収容可能なため
1GbE から **10GbE** サーバーへのシームレスな移行をサポート



※ EX3300、EX3400 では Mixed Mode Virtual Chassis はサポートされてません

中小エンタープライズ（450 ユーザ位まで）における キャンパスを一つの EX3400 Virtual Chassis で収容する例



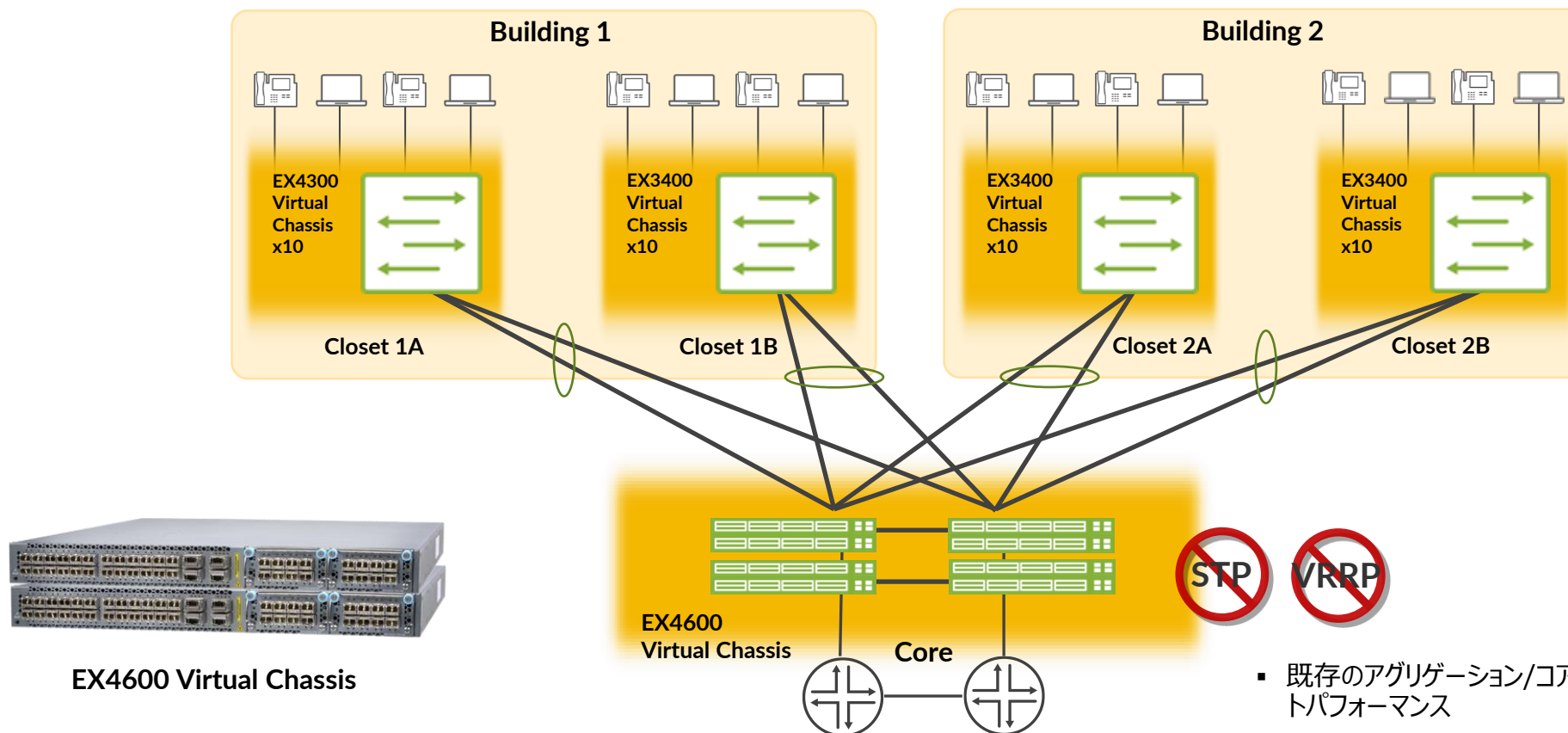
EX3400 Virtual Chassis



- キャンパス NW を 1 台のスイッチで収容するソリューション
- 最大 80km までを Virtual Chassis の 10GbE バックプレーンで収容
- STP を排除したネットワークデザイン

- Virtual Chassis を複数のワイヤリングクローゼット、ビル間で構築することで、
 - アップリンクポートの削減
 - 必要管理デバイス数の削減
- High Availability Virtual Chassis
- VRRP や複雑なルーティング、VLAN の管理が不要

EX4600 Virtual Chassis をキャンパスにおける Aggregation / Core スイッチとして使用する例

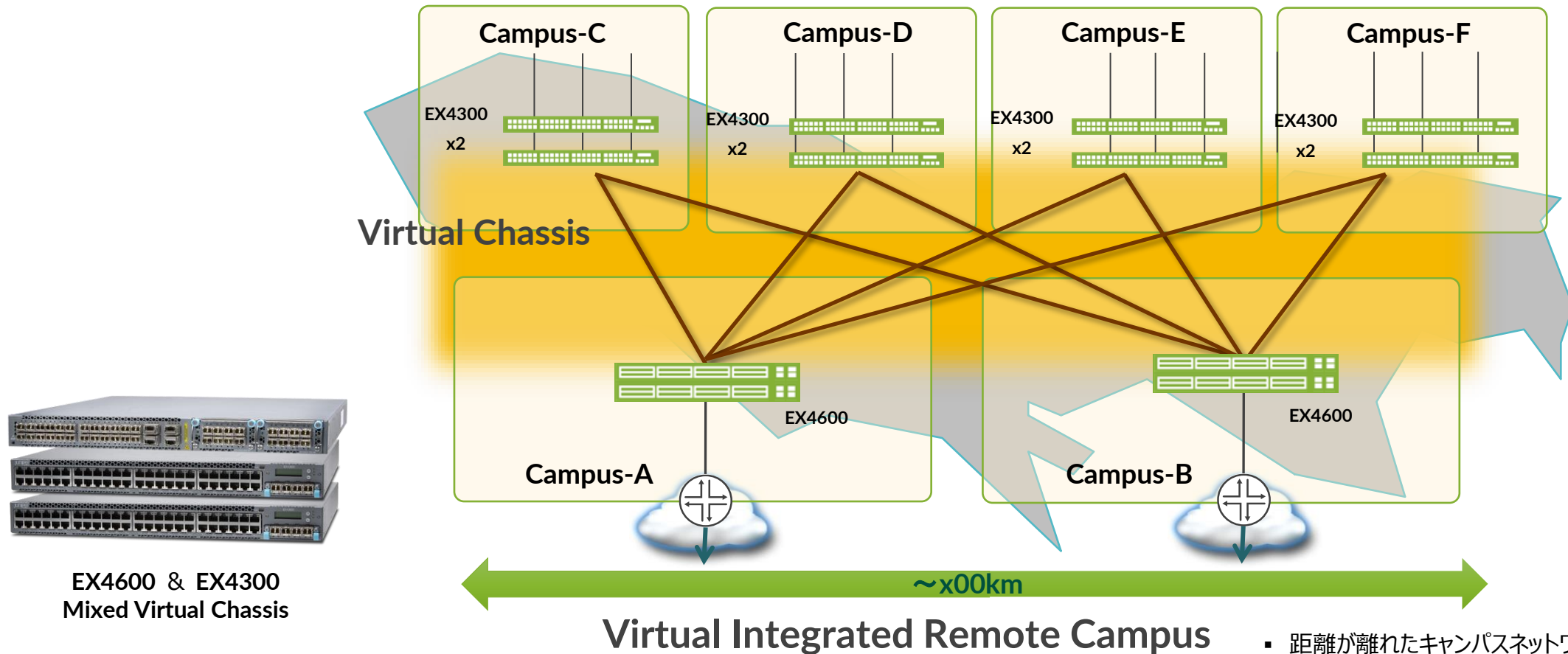


EX4600 Virtual Chassis

- 10GbE LAG によるワイヤリングクローゼットからのアップリンク
- 多数の EX3400 バーチャル・シャーシを冗長性を持って収容

- 既存のアグリゲーション/コア・スイッチと比較して圧倒的なコストパフォーマンス
 - コストは 1/8 に
 - パフォーマンスは 4 倍に
- STP を排除したネットワークデザイン
- VRRP や複雑なルーティング、VLAN の管理が不要

距離が離れたキャンパスネットワークを 1 セットの **Virtual Chassis** で収容する例



- 距離が離れたキャンパスネットワークを一台の仮想シャーシで集約することが可能

まとめ： Virtual Chassis によるメリット

最大で **10** 台のスイッチまでを
一つの仮想シャーシとして設定、管理運用が可能

Industry-
only



物理的に離れたデバイスであっても論理的に統合可能

Industry-
only



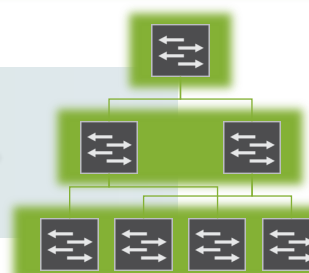
異なるプラットフォーム間でのバーチャルシャーシ接続
(e.g. QFX5100 + EX4300)

Industry-
only



コア、1G/10G/40G アクセス、マネジメント
規模やサービスレベルに応じた様々な VC を提供

Industry-
only



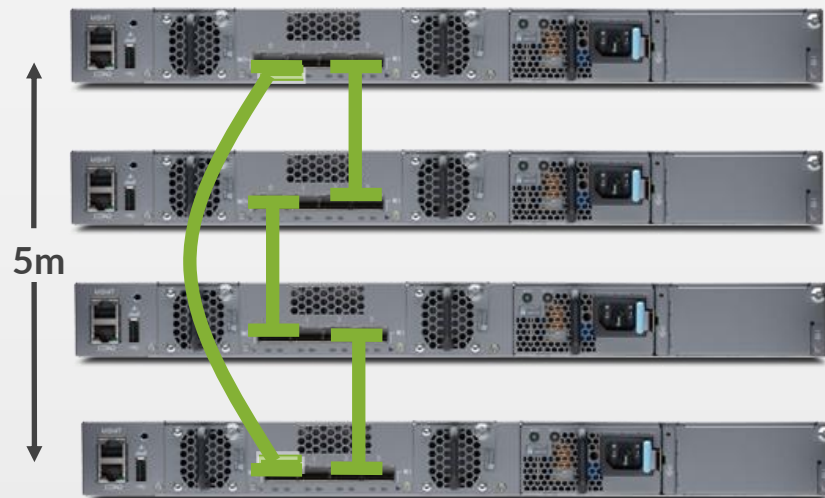
これにより拡張し続けるデータセンターのネットワークをシンプルに管理運用することが可能に！



Virtual Chassis Deep Dive

Virtual Chassis Backplane Cabling

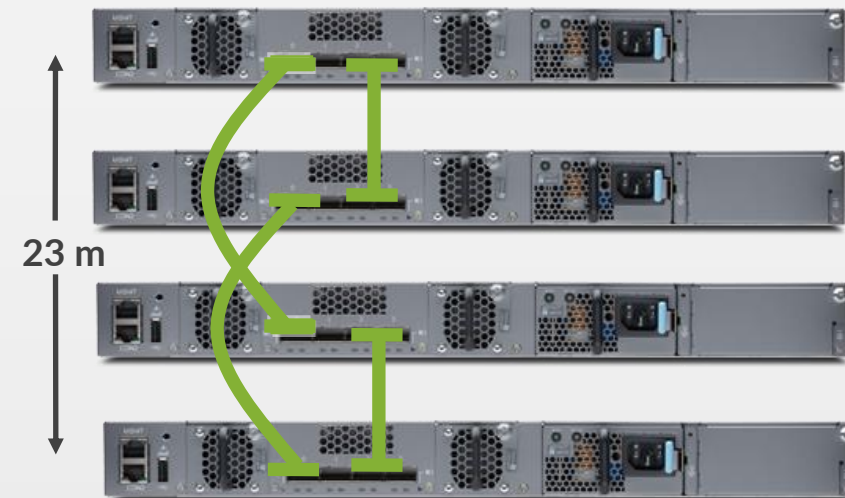
Option 1 – Dedicated Virtual Chassis Daisy-Chained Ring



Longest Virtual Chassis cable spans the entire Virtual Chassis

- もっともシンプルな接続方法
- VC の高さ・幅は VC ケーブルの最大長 5m 以内

Option 2 – Dedicated Virtual Chassis Braided Ring

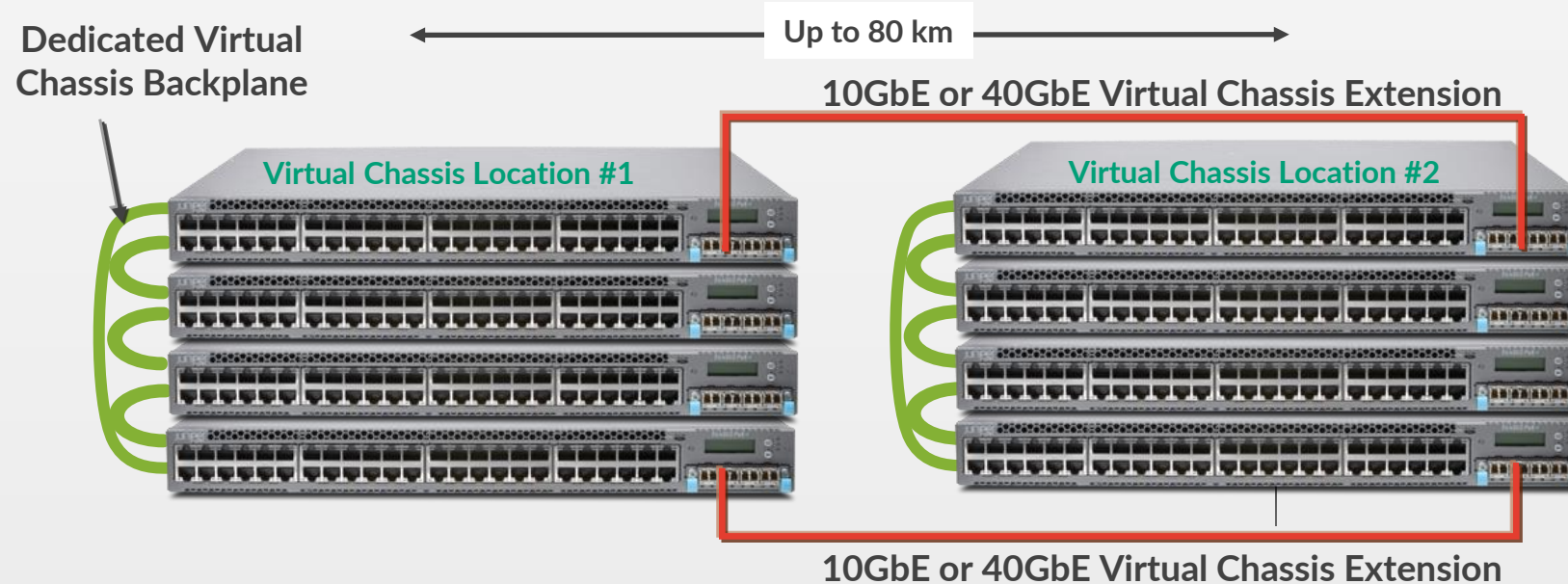


Longest Virtual Chassis cable spans three switches

- VC の高さ、幅を 約 23m まで拡張する接続方法

Virtual Chassis Backplane Cabling

Option 3 – Extended Virtual Chassis



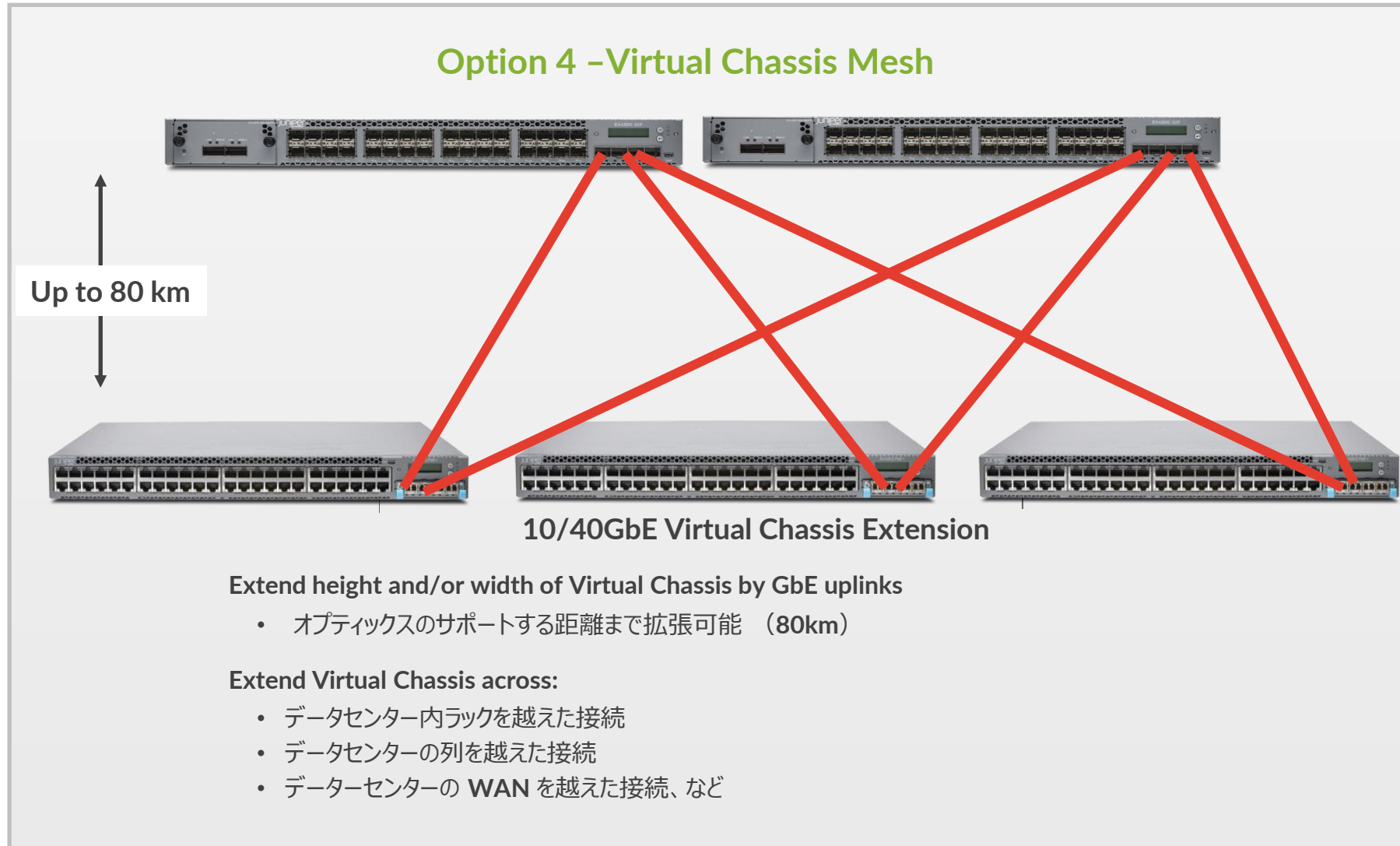
Extend height and/or width of Virtual Chassis by GbE or 10GbE uplinks

- オプティックスのサポートする距離まで拡張可能（70km）

Extend Virtual Chassis across:

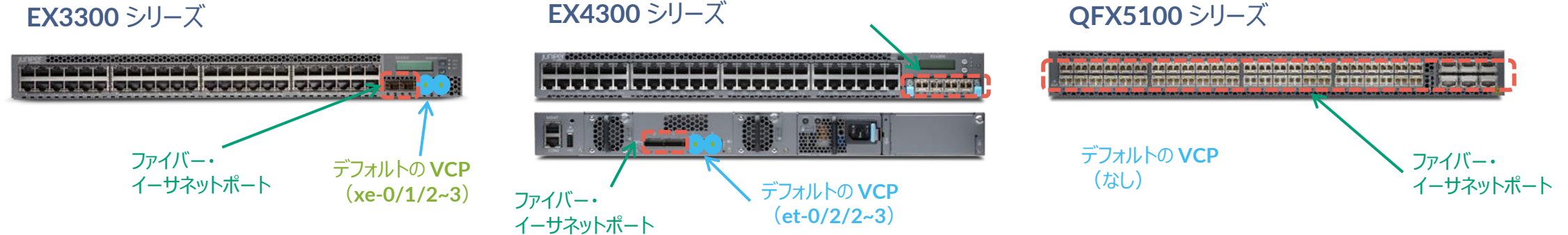
- ワイヤリングクローゼットを越えた接続
- データセンター内ラックを越えた接続
- データセンターの列を越えた接続、など

Virtual Chassis Backplane Cabling



Virtual Chassis の接続方法について - 1

VC を構成する際には、VC の仮想バックプレーン（VCP）同士を接続することが必要
プラットフォームによって工場出荷時の状態で VCP が設定されているものとそうでないものがある



ファイバーのイーサネットポートを VCP にコンバートするコマンドは以下で実行可能
必要に応じて VCP の設定追加・削除をした上で VC 接続を実施

```
request virtual-chassis vc-port set pic-slot <pic-slot> port <port-number> member <member-id>
request virtual-chassis vc-port delete pic-slot <pic-slot> port <port-number> member <member-id>
```

Virtual Chassis のコンポーネント

“Master”、“Backup” および “Linecard”

- Master switch (active RE)
- 相互接続された VC switch の 1つのスイッチがマスターになる
JUNOS を起動しており Virtual Chassis の管理を実施
 - すべての Virtual Chassis を管理するデーモンおよびコントロールプロトコルを動作させる
 - すべてのインタフェースを管理する、ハードウェアフォワーディングの管理を実施
- Backup switch (backup RE)
- 相互接続された VC switch の 1つのスイッチがバックアップになる
JUNOS を起動しておりバックアップとしてマスターと連携を実施
 - GRES 使用時は、RE0 とハードウェアフォワーディングテーブルを同期
 - RE0 が故障した場合に RE0 に変わり、シャーシの管理やインタフェース管理を実施
- Linecard switch (Linecard)
- その他のメンバーになっているスイッチはすべてラインカードになる
JUNOS を起動しておりラインカードとして動作
 - Non Preprovisioned Mode の場合、マスターかバックアップが故障した場合、ラインカードのひとつが新しいバックアップとして動作

Virtual Chassis の構成方法について - 1

VC を構成する際には、
Plug-and-Play での VC 構成を提供する “Non-Preprovisioned mode” と
最低限の設定投入により VC を構成する “Preprovisioned mode” から選択が可能

Non-preprovisioned Configuration

- マスター・セクション・アルゴリズムにより自動的に VC を構成することが可能
 - Master-ship priority 値や起動順序により、master / backup / linecard を決定
 - Master / BackupRE は、master-ship priority 255 を推奨
- RE の障害時には、Linecard 役の中から 1 台が RE に昇格する

Preprovisioned Configuration

- 明示的に RE や Linecard に指定したスイッチを作成することで、より明示的な運用の実現と Advanced License の消費を抑えることが可能
 - ※ NSSU は Preprovisioned Configuration でのみサポート

Virtual Chassis の構成方法について - 2

より簡易性が求められるネットワークへのデプロイ時には “**Non-Preprovisioned mode**” で VC を構成
“**Non-Preprovisioned mode**” では予め設定されたルールに基づき、どの筐体が **Routing Engine** の役割を担うか自動的に計算されて VC が構成される

- **マスター RE (RE0) 選定**

起動するときにはすべてのスイッチで以下項目比較の元、マスターの選定が行われる

Master 選定の優先順位:

1. マスターシップの優先順位が最も高い (0 – 255 までの優先順位、デフォルト値は 128)

```
> set virtual-chassis member <member-id> mastership-priority <priority 1-255>
```

2. 以前動作していたときにマスターに選定されていた
3. 起動している時間が長い (起動している時間が 1 分以上違う場合)
4. **MAC** アドレスの小さいほう

※マスターが選定された後、マスター RE と同じ選定方式により、バックアップ RE スwitchの選定が実施される

- **Linecard**

バーチャル・シャーシを構成する残りのスイッチは、ラインカードとして動作

マスター、バックアップが何らかの理由によりフェイルした場合、マスター RE と同じ選定方式によりラインカードからバックアップスイッチの選定を実施

Virtual Chassis の構成方法について - 3

より高い **SLA** が求められるネットワークへのデプロイ時には “**Preprovisioned mode**” での **VC** 構成が推奨となる

“**Preprovisioned mode**” では設定によりシリアルでのハードウェアと **Role** 管理によるより安定した運用と、**OS** アップグレード時にミニマムなダウンタイムでの実施完了を期待できる **NSSU**（**Non Stop Software Upgrade**）サービスが提供される

```
set virtual-chassis preprovisioned
set virtual-chassis member 0 role routing-engine
set virtual-chassis member 0 serial-number 111111111111
set virtual-chassis member 1 role line-card
set virtual-chassis member 1 serial-number 222222222222
set virtual-chassis member 2 role line-card
set virtual-chassis member 2 serial-number 333333333333
set virtual-chassis member 3 role routing-engine
set virtual-chassis member 3 serial-number 444444444444
```

Preprovisioned mode を宣言

各筐体毎のシリアル No. を投入

任意の筐体 2 台で **Routing-Engine Role** を宣言、
その他の筐体の **Role** はすべて **Linecard**

Virtual Chassis の確認方法について - 1

lab@lab> show virtual-chassis ?

```
root@Juniper> show virtual-chassis ?
Possible completions:
  <[Enter]>           Execute this command
  active-topology     Virtual chassis active topology
  device-topology     PFE device topology
  login
  mode                Virtual chassis mode information
  protocol            Show virtual chassis protocol information
  status              Virtual chassis information
  vc-path             Show virtual-chassis packet path
  vc-port             Virtual chassis port information
  |                  Pipe through a command
{master:0}
```

Virtual Chassis の確認方法について - 2

“show virtual-chassis status” コマンドにて構成された VC の状態を確認することが可能です

```
root> show virtual-chassis status
```

Virtual Chassis ID: 0019.e255.3740

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	BM0208124253	ex4200-24t	128	Master*	1	vcp-0
						2	vcp-1
1 (FPC 1)	Prsnt	BM0208124327	ex4200-24t	128	Backup	2	vcp-0
						0	vcp-1
2 (FPC 2)	Prsnt	BM0208124235	ex4200-24t	128	Linecard	0	vcp-0
						1	vcp-1

Member ID for next new member: 3 (FPC 3)

Non-Preprovisioned Mode の場合、Default ではすべてのメンバーの Mastership Priority は 128 となる (RE はマニュアルで 255 に変更することを推奨)

```
root> show virtual-chassis status
```

Preprovisioned Virtual Chassis

Virtual Chassis ID: 0019.e255.3740

Member ID	Status	Serial No	Model	Mastership priority	Role	Neighbor List ID	Interface
0 (FPC 0)	Prsnt	BM0208124253	ex4200-24t	129	Master*	1	vcp-0
						2	vcp-1
1 (FPC 1)	Prsnt	BM0208124327	ex4200-24t	129	Backup	2	vcp-0
						0	vcp-1
2 (FPC 2)	Prsnt	BM0208124235	ex4200-24t	0	Linecard	0	vcp-0
						1	vcp-1

Preprovisioned Mode の場合、RE の Mastership Priority が 129 となり、Linecard の Mastership Priority は 0 となる

Virtual Chassis の確認方法について - 3

“show virtual-chassis vc-port” コマンドにて VC バックプレーンの状態を確認することが可能です

```
root@Juniper> show virtual-chassis vc-port
fpc0:
-----
Interface      Type           Trunk  Status      Speed      Neighbor
or             PIC / Port      ID      (mbps)      ID  Interface
vcp-0          Dedicated       2      Up          32000      1    vcp-1
vcp-1          Dedicated       1      Up          32000      9    vcp-0
1/0            Configured      -1      Up          1000       1    vcp-255/1/1
1/1            Configured      -1      Up          1000       3    vcp-255/1/0

fpc1:
-----
Interface      Type           Trunk  Status      Speed      Neighbor
or             PIC / Port      ID      (mbps)      ID  Interface
vcp-0          Dedicated       2      Up          32000      2    vcp-1
vcp-1          Dedicated       1      Up          32000      0    vcp-0
1/0            Configured      -1      Up          1000       1    vcp-255/1/1
1/1            Configured      -1      Up          1000       0    vcp-255/1/0

fpc2:
-----
...
```

Virtual Chassis Mixed Mode

QFX5100、EX4600、EX4300 などにおいては、異なるメディアのプラットフォームを 1 台の VC として構成させることも可能

その場合、VC に組み込む前に以下のコマンドで VC の **Mixed Mode** を宣言して機器を **Reboot** することが必要（VC を構成するメンバー全ての筐体で宣言することが必要）

```
request virtual-chassis mode mixed
request system reboot
```



QFX5100



EX4600



EX4300

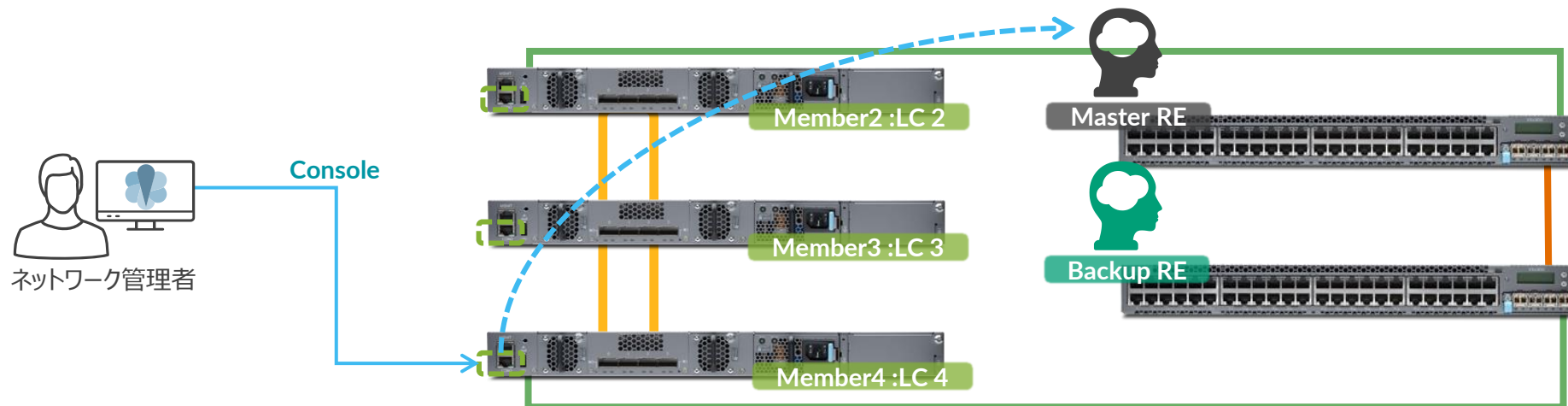
Virtual Chassis へのアクセスについて - 1

Virtual Chassis を構成すると、複数台のスイッチが 1 台の仮想シャーシ型スイッチとして動作

VC へのアクセスは **Console** 接続経由とネットワーク経由と二種類の選択肢がありますが、それぞれ以下の様な概念で動作

- コンソールアクセス

ネットワーク管理者は任意のラインカード上のコンソールポートに接続すると、接続コネクションが内部的に **Master RE** にリダイレクトされる（物理的な場所を気にする必要なく **RE** にアクセスすることが可能）



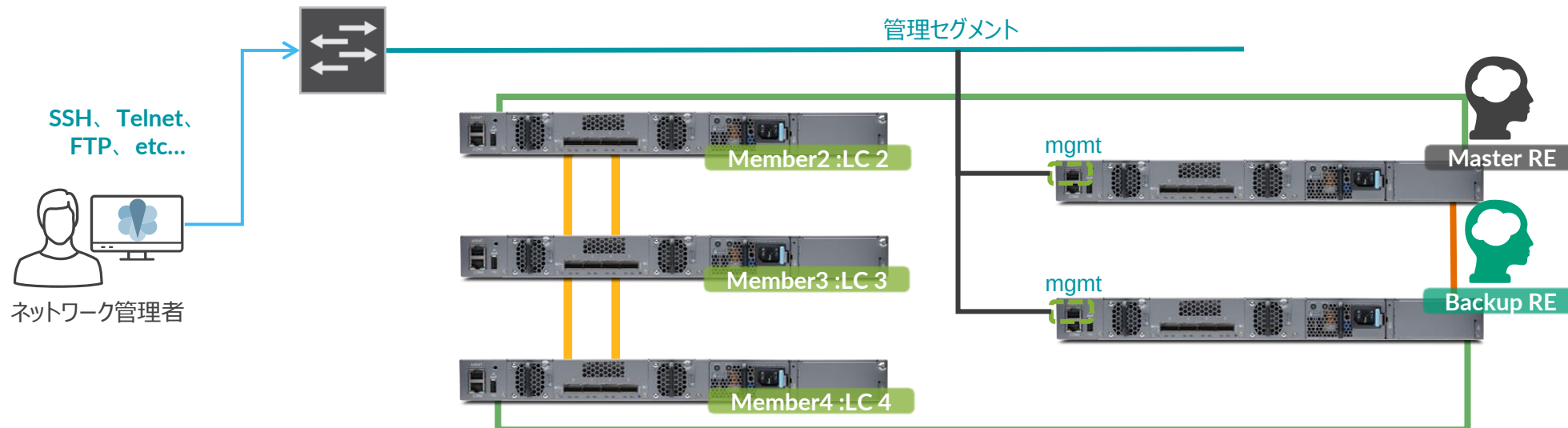
Virtual Chassis へのアクセスについて - 2

- ネットワークアクセス

仮想管理アドレスである **VME**（**Virtual Management Ethernet**）に IP アドレスを付与することで **Master RE** が **VME** アドレスへのアクセス要求に返答を行う
これによりひとつの **IP** アドレスで仮想シャーシへのネットワークアクセスが提供される

```
set interface vme unit 0 family inet address <address/mask>
```

ケーブリングは **Master RE** になりうる 2 つの筐体でのみリンクアップさせておけば他は不要



Virtual Chassis の HA 機能について

Routing Engine (RE) の障害時に出来る限り高速な切り替わりを提供するためには、以下の 4 行の設定投入をしておくことが必要
VC の初期構成時点で使用している L2 / L3 プロトコルの種類に限らずこの 4 行の設定は無条件に投入しておくことが推奨となる

EX3400 / 4300 / EX4600 / QFX5100 シリーズ

```
set chassis redundancy graceful-switchover
set routing-options nonstop-routing
set protocols layer2-control nonstop-bridging
set system commit synchronize
```

Kernel や Interface、L2 / L3 テーブルを RE 間で同期

L3 のプロトコルステータスを RE 間で同期

L2 のプロトコルステータスを RE 間で同期

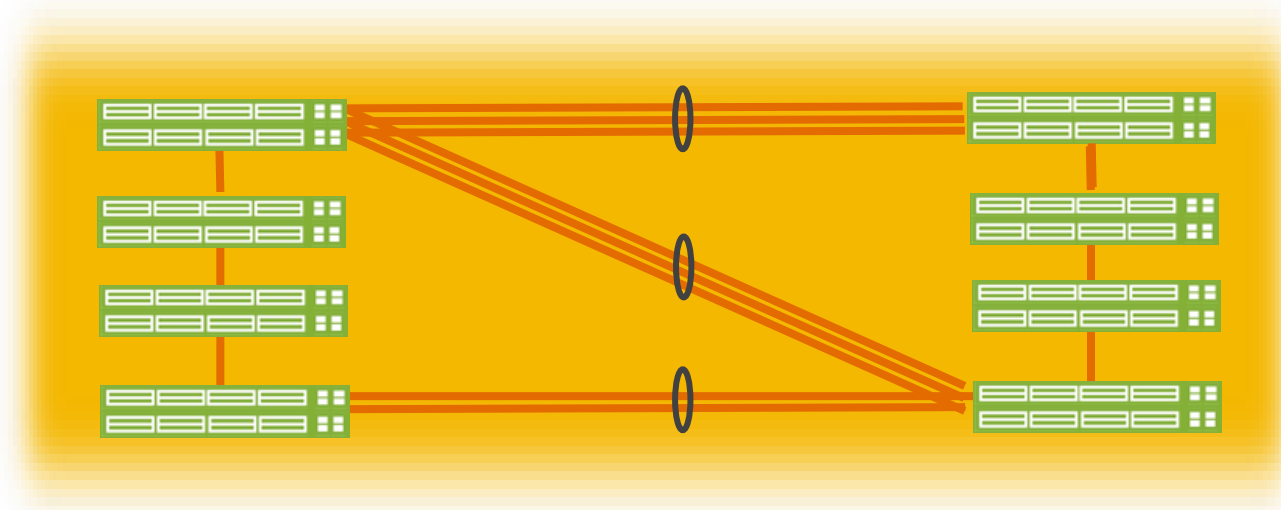
おまじない

EX3300 シリーズ

```
set chassis redundancy graceful-switchover
set routing-options nonstop-routing
set ethernet-switching-options nonstop-bridging
set system commit synchronize
```


Virtual Chassis バックプレーン増強について

同一 VC メンバー間で複数の仮想バックプレーン（VCP）が接続されたことを VC が認識すると、その間は自動的に LAG が構成され、バックプレーン帯域がリンク数 * N へと増強される（設定は不要）



Virtual Chassis に関するドキュメント

以下に **Virtual Chassis** を解説する各種資料がありますので、必要に応じてご参照ください

- Links

https://www.juniper.net/techpubs/en_US/junos14.1/information-products/pathway-pages/qfx-series/virtual-chassis.pdf

https://www.juniper.net/techpubs/en_US/junos14.1/topics/concept/virtual-chassis-ex-qfx-series-mixed-understanding.html

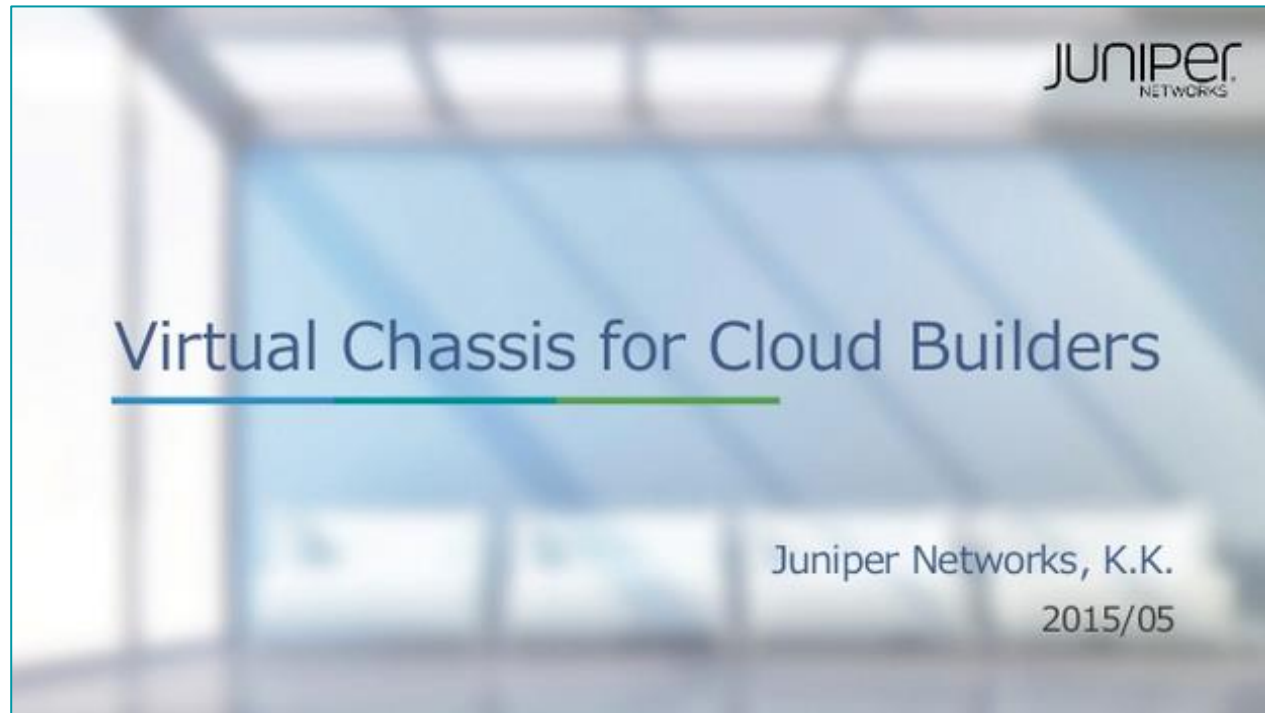
- Whitepaper

<http://www.juniper.net/us/en/local/pdf/whitepapers/2000427-en.pdf>

Virtual Chassis に関するドキュメント

Virtual Chassis for Cloud Builders

<http://www.slideshare.net/JuniperJapan/vc4-cb-201505>



Virtual Chassis の使い方や内部動作詳細を日本語で解説！

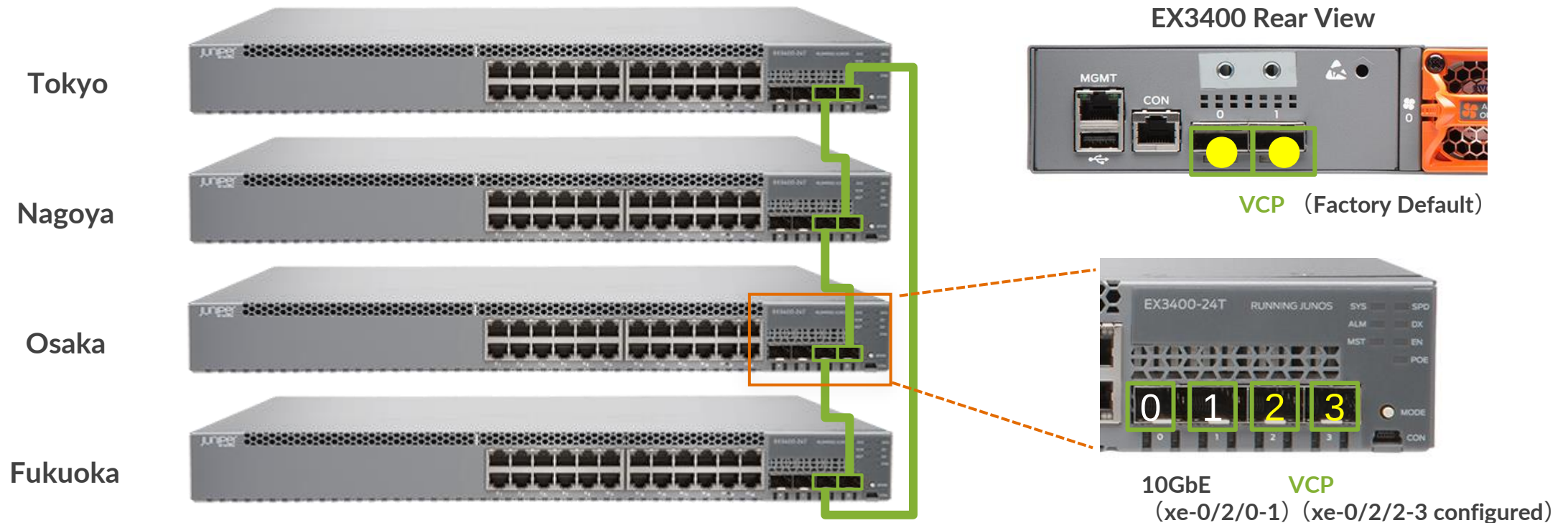


LAB.5

Virtual Chassis の設定

Ethernet Switching “EX/QFX” Course

Topology (Lab.5 : Virtual Chassis を構成)



※ EX3400 シリーズは工場出荷状態では 2 つの 40GbE インタフェースが **VC Port** としてデフォルトで設定
本トレーニングではケーブルの都合上、4 つの 10GbE インタフェースのうち老番の 2 ポートを **VC Port** として事前設定

VC 事前確認 ①

VC を構成する前に、単体の EX で以下を事前に確認

- Member ID が “0” であること
- Member ID for next new member が “1” であること
- Mixed Mode が “N” または “NA” となっていること
- Master priority が “128” であること

```
> show virtual-chassis
Virtual Chassis ID: 2d90.26d8.22f2
Virtual Chassis Mode: Enabled

Member ID  Status  Serial No  Model  Mstr  Role  Mixed Neighbor List
           (FPC 0)  Prsnt     BR0208392392  ex4200-24t  prio  Master*  Mode ID  Interface
0 (FPC 0)  Prsnt     BR0208392392  ex4200-24t  128   Master*  N

Member ID for next new member: 1 (FPC 1)
```

- Config 上に virtual-chassis に関連する設定が何も入っていないこと

```
# show virtual-chassis
```

- EX の Junos SW バージョンが他のメンバーと同一であること

VC 事前確認 ② EX3400

VC に使用するポートの設定を確認

- 2 つのポート 2/2 、 2/3 が VC Port に設定されていること
- Type が Configured と表示されていること
- Status が Absent 、または Down であること

```
{master:0}
lab@EX3400> show virtual-chassis vc-port
fpc0:
-----
Interface      Type              Trunk   Status      Speed      Neighbor
or              ID               ID      (mbps)      ID  Interface
PIC / Port
2/3             Configured                Absent
2/2             Configured                Absent
1/0             Configured                Absent
1/1             Configured                Absent
```


① VC Basic Setup (non pre-provisioned)

- Tokyo 以外のスイッチで、電源を OFF

```
root> request system halt at now
```

- (Tokyo のみ) VC 管理用インタフェースとして me0 の設定を vme に変更

```
# rename interfaces me0 to vme  
# commit
```

- Tokyo と Nagoya の VC Port を接続し、Nagoya の電源を ON
Tokyo が Master に選定され、Tokyo の IP アドレスに再接続
- 同様に、Osaka 、Fukuoka をそれぞれ順に接続し、電源を起動

① VC 基本構成確認

- 以下のコマンドで VC のステータスを確認

```
> show virtual-chassis status  
> show virtual-chassis vc-port  
> show virtual-chassis login
```

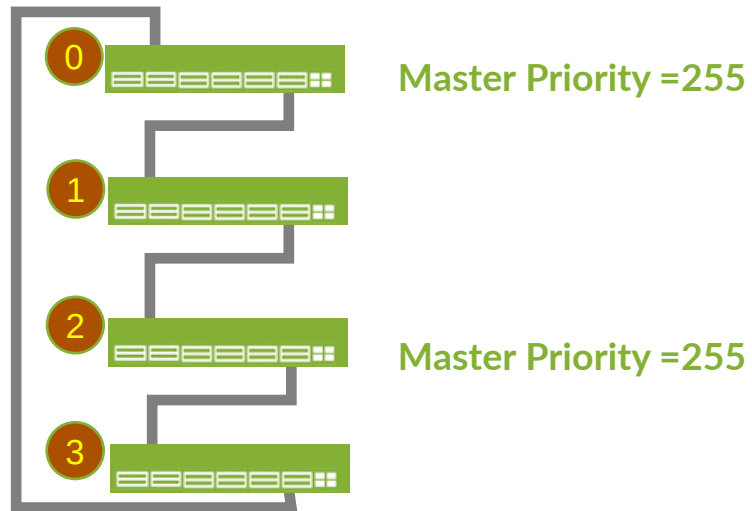
- 以下のコマンドで VC メンバーの機器にログイン

```
> request session member <member-id>
```

② mastership priority の変更 (non pre-provisioned)

- Mastership Priority を変更し、任意の EX を Master RE 、 Backup RE に指定
- 以下の設定を Master となっているスイッチで実行

```
root# set virtual-chassis member 0 mastership-priority 255
root# set virtual-chassis member 2 mastership-priority 255
root# commit synchronize
```



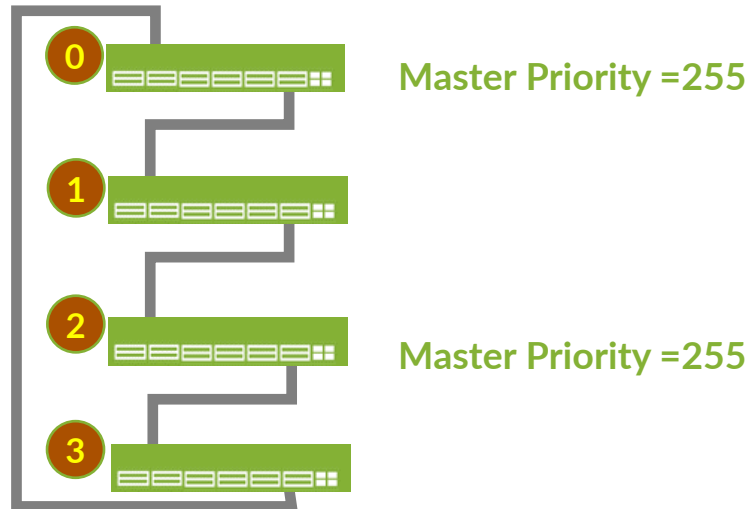
② mastership priority の変更 (non pre-provisioned)

- Mastership Priority が変更され、ステータスが更新されたことを確認

```
root> show virtual-chassis
root> show virtual-chassis vc-port statistics
```

- Tokyo と Nagoya 間の VC ケーブルを抜去し、ステータスや VC ポートの遷移を確認

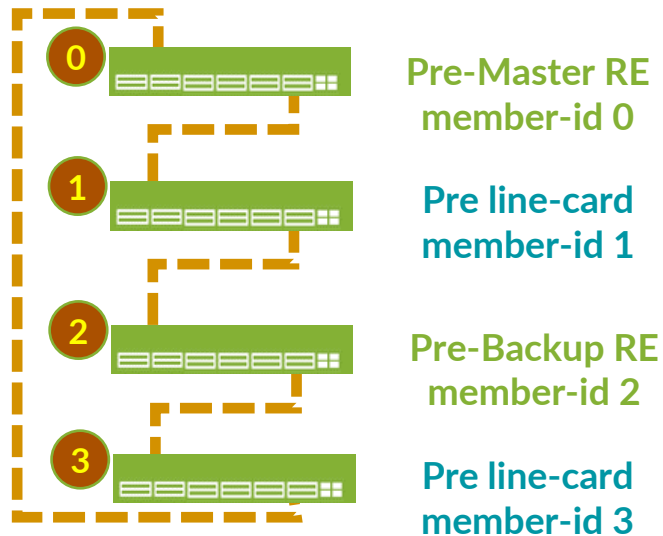
```
root> show virtual-chassis
```



③ Virtual Chassis のリセット

- VC を解体して、4 台の個別なスイッチに戻す
- VCP ケーブルを抜き、各スイッチでステータスを確認
- 解体後、以下コマンドを実行し、各種 VC 情報を消去（※ 以下は Backup-RE の例）

```
root> request virtual-chassis reactivate
root> request virtual-chassis recycle member-id 0
root> request virtual-chassis recycle member-id 1
root> request virtual-chassis recycle member-id 3
root> request virtual-chassis renumber member-id 1 new-member-id 0
```



Virtual Chassis ハンズオン 応用編

※ 説明のみ

※ ④、⑤ の手順は、本コースのハンズオンでは実施しません

④ Pre-provisioned configuration での VC 設定

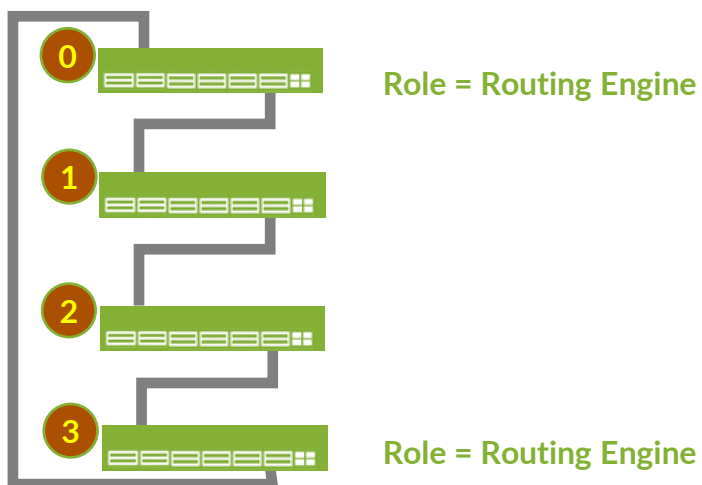
⑤ Virtual Chassis HA (Virtual Chassis 冗長構成の設定)

④ Pre-provisioned configuration

※ 説明のみ

- VC を Pre-provision Configuration で構成
- Master RE に以下の設定を投入後、各メンバーの VC ポートを接続

```
root# set virtual-chassis preprovisioned  
root# set virtual-chassis member 0 role routing-engine  
root# set virtual-chassis member 0 serial-number xxxxxxxxxxxx  
root# set virtual-chassis member 1 role line-card  
root# set virtual-chassis member 1 serial-number xxxxxxxxxxxx  
root# set virtual-chassis member 2 role line-card  
root# set virtual-chassis member 2 serial-number xxxxxxxxxxxx  
root# set virtual-chassis member 3 role routing-engine  
root# set virtual-chassis member 3 serial-number xxxxxxxxxxxx
```

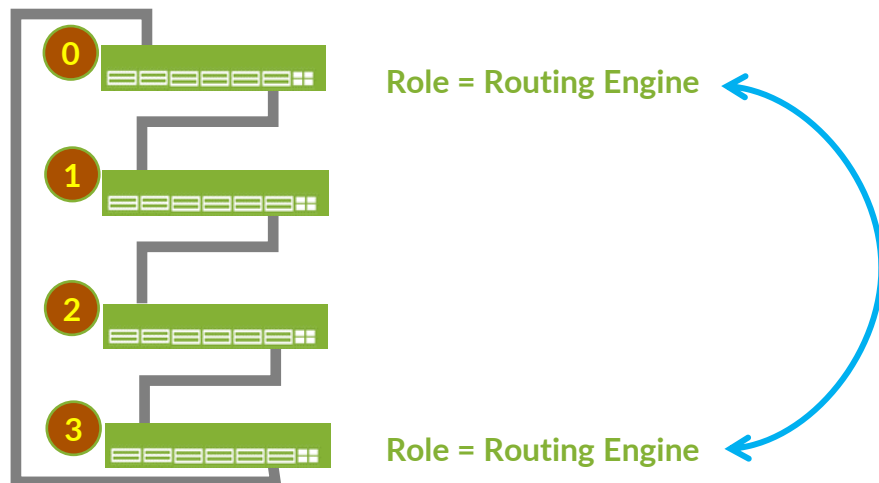


⑤ Virtual Chassis HA

※ 説明のみ

- RE 間のテーブルやプロトコルの同期設定を行うことで、RE 障害のダウンタイムを軽減する

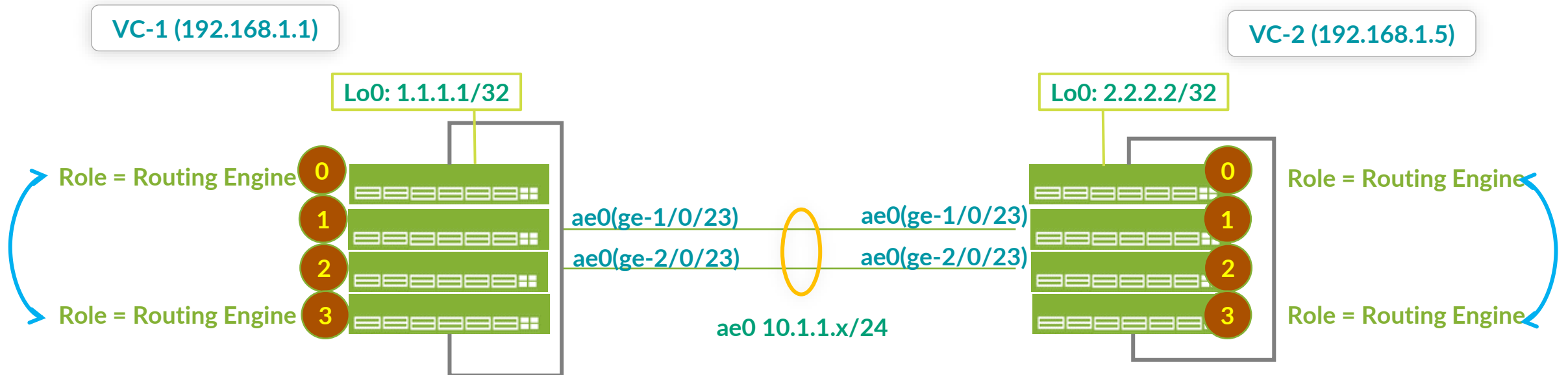
```
root# set chassis redundancy graceful-switchover
root# set routing-options nonstop-routing
root# set protocols layer2-control nonstop-bridging
root# set system commit synchronize
```



⑤ Virtual Chassis HA

※ 説明のみ

- OSPF の設定を VC-1 ・ VC-2 に投入した後に、Master RE を Halt して NSR の効果を確認



⑤ Virtual Chassis HA

※ 説明のみ

- 設定サンプル

VC-1(192.168.1.1)

```
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-1/0/23 ether-options 802.3ad ae0
set interfaces ge-2/0/23 ether-options 802.3ad ae0
set interfaces ae0 unit 0 family inet address 10.1.1.1/24
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set routing-options router-id 1.1.1.1
set protocols ospf area 0.0.0.0 interface ae0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

VC-2(192.168.1.5)

```
set chassis aggregated-devices ethernet device-count 1
set interfaces ge-1/0/23 ether-options 802.3ad ae0
set interfaces ge-2/0/23 ether-options 802.3ad ae0
set interfaces ae0 unit 0 family inet address 10.1.1.2/24
set interfaces lo0 unit 0 family inet address 2.2.2.2/32
set routing-options router-id 2.2.2.2
set protocols ospf area 0.0.0.0 interface ae0.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

⑤ Virtual Chassis HA

※ 説明のみ

- 確認方法

(VC-1 の Master RE を再起動し、対向の VC-2 側の OSPF neighbor が切れないことを確認)

VC-1(192.168.1.1)

```
{master:0}[edit]
root@Tokyo-1# run request system reboot member 0 at now
Reboot the system at now? [yes,no] (no) yes
```

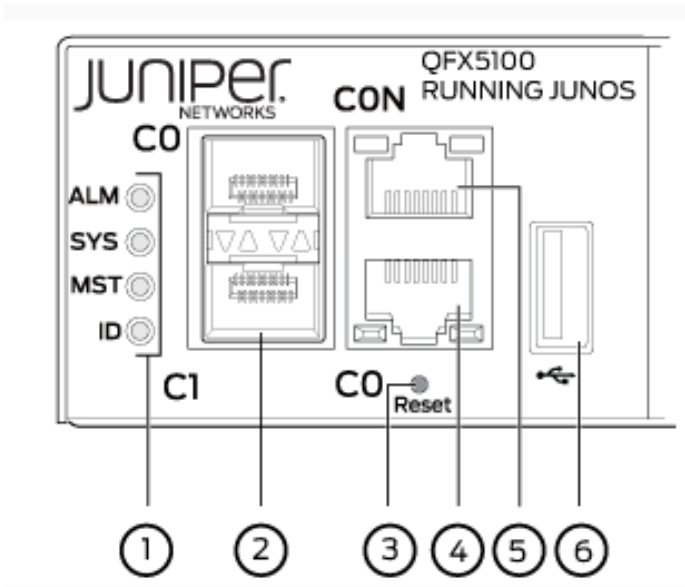
VC-2(192.168.1.5)

```
{master:0}[edit]
root@Tokyo-2# run show ospf neighbor detail
Address          Interface          State      ID                Pri   Dead
10.1.1.1          ae0.0              Full       1.1.1.1           128   38
Area 0.0.0.0, opt 0x52, DR 10.1.1.2, BDR 10.1.1.1
Up 00:03:24, adjacent 00:03:24
```

Virtual Chassis の目視での確認方法

- Virtual Chassis の状態は **Status LED** を目視することで状態の確認を行うことが可能

QFX5100 の場合



SYS (System)

- 消灯：システムがパワーオフ、もしくは **Halt** 状態
- 点灯： **Junos** がスイッチ上で動作している状態
- 点滅： **Virtual Chassis (VC)** のメンバースイッチ

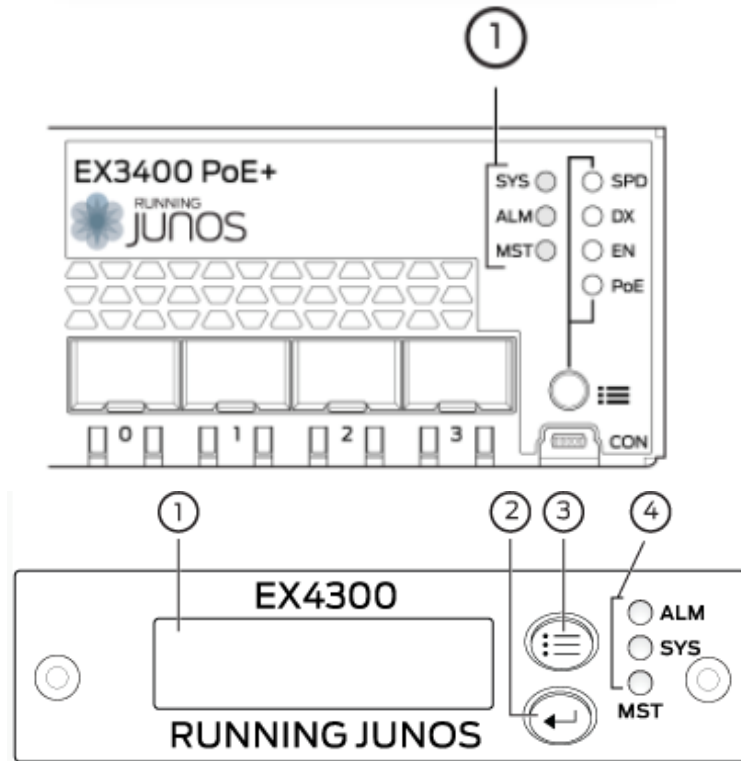
MST (Master)

- 消灯： **VC** の **Linecard** として動作
- 点灯： 以下のうちどれかの状態
 - スタンドアローンスイッチ
 - VC** の **Master RE** スイッチ
- 点滅： **VC** の **Backup RE** スイッチ

Virtual Chassis の目視での確認方法

- Virtual Chassis の状態は **Status LED** を目視することで状態の確認を行うことが可能

EX3400 / EX4300 の場合



SYS (System)

- 点灯： **Junos** がスイッチ上で動作している状態
- 点滅： スイッチが起動中の状態
- 消灯： システムがパワーオフ、もしくは **Halt** 状態

MST (Master)

- Standalone** の場合
 - 消灯： システムがパワーオフ、もしくは **Halt** 状態
 - 点灯： **Junos** がスイッチ上で動作している状態
- Virtual Chassis** の場合
 - 点灯： **VC** の **Master RE** スイッチ
 - 点滅： **VC** の **Backup RE** スイッチ
 - 消灯： **VC** の **Linecard** スイッチ、もしくは **Halt** 状態



THANK YOU

JUNIPER
NETWORKS | Driven by
Experience™



Appendix

- A. TIPs to be Junos Experts
- B. Multi-Chassis LAG
- C. Zero Touch Provisioning



Appendix A: TIPs to be Junos Experts

俳句の表示

- 検証作業やトラブルシュートに疲れたときには、**Junos** に前向きな気持ちの言葉を表示させ、管理者の気持ちを和らげることが可能

```
root> show version and haiku
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package [14.1X53-D25.2]
```

```
Look, mama, no hands!
Only one finger typing.
Easy: commit scripts.
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package [14.1X53-D25.2]
```

```
Juniper babies
The next generation starts
Gotta get more sleep
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package [14.1X53-D25.2]
```

```
Weeks of studying,
Days of lab exercises:
JNCIE.
```

※コマンドを打つ度、異なった前向きなポエムが表示される

設定のコピー

- **copy** コマンドにより特定の設定をコピーすることが可能

ge-0/0/1 の設定を ge-0/0/0 へコピー

```
root# copy interfaces ge-0/0/1 to ge-0/0/0
```

```
root# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
```



```
root# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
```


設定の書き換え

- **rename** コマンドにより設定した **variable** やエレメントを書き換えることも可能
ge-0/0/0 の address を 192.168.2.1/26 へ変更

```
root# rename interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/26 to address 192.168.2.1/26
```

```
root# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
    }
  }
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
    }
  }
}
```



```
root# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.2.1/26;
    }
  }
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/26;
    }
  }
}
```

設定の項目の置換

- **replace** コマンドにより設定内の文字列を置換することも可能

ge-0/0/0 の address を 192.168.2.1/26 へ変更

```
root# replace pattern /26 with /24
```

```
root# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.2.1/26;
        }
    }
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.1.1/26;
        }
    }
}
```



```
root# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.168.2.1/24;
        }
    }
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.168.1.1/24;
        }
    }
}
```

activate / deactivate

- **deactivate** コマンドを使うことで、設定の一部を削除することなく無効にすることが可能なので、障害時の切り分けなどに便利

192.168.1.2/24 を無効化

```
root# deactivate interfaces ge-0/0/1 unit 0 family inet address 192.168.1.2/24
```

```
root# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/24;
      address 192.168.1.2/24;
```



```
root# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      address 192.168.1.1/24;
      inactive: address 192.168.1.2/24;
```

192.168.1.2/24 の無効化を解除（有効化）

```
root# activate interfaces ge-0/0/1 unit 0 family inet address 192.168.1.2/24
```


wildcard range set / delete

- **wildcard range** コマンドを使用することで、インターフェイスなど複数の対象に対して同じ設定内容を適用することが簡単に可能

```
root# show interfaces
```

```
root#
```

```
root# wildcard range set interfaces ge-0/0/[0-3,5,!2] mtu 9000
```

[0-3, 5, !2] ⇒ 0 ~ 3 と 5、ただし 2 は除く



```
root# show interfaces
ge-0/0/0 { mtu 9000; }
ge-0/0/1 { mtu 9000; }
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```

ge-0/0/0、1、3、5 の MTU 設定が一括で投入されている

wildcard range set / delete

- 同様に delete も可能

```
root# show interfaces
ge-0/0/0 { mtu 9000; }
ge-0/0/1 { mtu 9000; }
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```

```
root# wildcard range delete interfaces ge-0/0/[0-1] mtu
```



```
root# show interfaces
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```

interface-range

- **interface-range** を使用することで、複数のインターフェイスをグループ化して共通の設定を行う事が可能。この設定は **wildcard** と異なりコンフィグ内に保持される為、一度作成してしまえば様々な設定に対する繰り返しの利用が可能

```
root# show interfaces
```

```
root#
```

```
root# set interfaces interface-range CLIENTS member-range ge-0/0/0 to ge-0/0/1
```

```
root# set interfaces interface-range CLIENTS member ge-0/0/3
```

```
root# set interfaces interface-range CLIENTS mtu 9000
```



clients というメンバーに入っている、**ge-0/0/0-1、3** の **MTU** を一括設定

```
root# show interfaces
interface-range CLIENTS {
    member ge-0/0/3;
    member-range ge-0/0/0 to ge-0/0/1;
    mtu 9000;
}
```

interface-range

- **range** 内の個別インターフェイス毎に特有の設定を追加することも可能

```
root# show interfaces
interface-range CLIENTS {
  member ge-0/0/3;
  member-range ge-0/0/0 to ge-0/0/1;
  mtu 9000;
}
```

```
root# set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/24
```



clients というメンバー共通でない
設定を IF 単体に設定設定

```
root# show interfaces
interface-range clients {
  member ge-0/0/3;
  member-range ge-0/0/0 to ge-0/0/1;
  mtu 9000;
}
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.0.0.1/24;
    }
  }
}
```

階層間の移動 - 1

同じ階層の設定を複数作成する際は階層を移動することで作成する構文を省略することが可能

- 例 1 : FW フィルタの設定 (top の階層から設定)

```
# show firewall
family inet{
  filter FW-FILTER{
    term BLOCK{
      from{
        source-address{
          10.10.10.0/24;
        }
        destination-address{
          192.168.1.0/24;
        }
        dscp cs5;
        port[ https http ];
      }
    }
  }
}
```



```
[edit]
set firewall family inet filter FW-FILTER term BLOCK from
source-address 10.10.10.0/24
set firewall family inet filter FW-FILTER term BLOCK from
destination-address 192.168.1.0/24
set firewall family inet filter FW-FILTER term BLOCK from
dscp cs5
set firewall family inet filter FW-FILTER term BLOCK from
port https
set firewall family inet filter FW-FILTER term BLOCK from
port http
```

※設定を投入する際は**繰り返し** set firewall family...from と入力することが必要

階層間の移動 - 2

例 2 : FW フィルタの設定 (firewall filter FW-FILTER term BLOCK from の階層から設定)

```
# show firewall
family inet {
  filter FW-FILTER {
    term BLOCK {
      from {
        source-address {
          10.10.10.0/24;
        }
        destination-address {
          192.168.1.0/24;
        }
        dscp cs5;
        port [ https http ];
      }
    }
  }
}
```



```
[edit firewall family inet filter FW-FILTER term BLOCK from]
set source-address 10.10.10.0/24
set destination-address 192.168.1.0/24
set dscp cs5
set port https
set from port http
```

※設定を投入する際は **firewall family...from** までを**省略して**入力することが可能

階層間の移動 - 3

- 階層間は、**edit** コマンドで移動することが可能
- **exit** : 直前にいたレベルに戻る
 - **top** で **exit** を実行すると、**Operational** モードに戻る
 - **Operational** モードで **exit** を実行すると、システムから **Logout**
 - **Shell** モードから 'cli' で **Operational** モードに移動した場合は、**Shell** モードに戻る
- **up** : 一つ上のレベルに移動
- **top** : 最上位のレベルに移動

edit
で階層を指定
↓

top
で最上位へ
↑

up
で一つ上へ
↑

Top
↑
Down



```
# show firewall
family inet{
  filter FW-FILTER{
    term BLOCK{
      from{
        source-address{
          10.10.10.0/24;
        }
        destination-address{
          192.168.1.0/24;
        }
        dscp cs5;
        port[ https http ];
      }
    }
  }
}
```


Automatic Configuration Archival

- **Automatic Configuration Archival** 機能を使用することで、自動的に最新のコンフィグをリモートの **FTP / SCP** サーバにバックアップすることが可能
- アップロードのタイミングは、コミットの度もしくは一定時間毎のいずれか、あるいは両方を選択可能

1. コミットの度にリモートのサーバにコンフィグをバックアップする設定：

```
user@Junos# set system archival configuration transfer-on-commit
user@Junos# set system archival configuration archive-sites ftp://
loginname:loginpassword@FTP-server-ip/directory
```

2. 一定時間おきにリモートのサーバにコンフィグをバックアップする設定： (例： 1440 分 = 24 時間おき)

```
user@Junos# set system archival configuration transfer-interval 1440
user@Junos# set system archival configuration archive-sites ftp://
loginname:loginpassword@FTP-server-ip/directory
```

機器の初期化

- **Junos 機器を初期化する手法は主に以下の 3 つ**
 - **Configuration mode で load factory-default**
 - 実行すると、**Candidate Configuration** にデフォルトの設定がロードされる
 - 実際に初期設定に戻すには、**root** パスワードの設定と **Commit** が必要
 - 設定のみを戻したいときに有効で、ログや過去の **Config (rollback)** などは削除されない
 - **Operation mode で request system zeroize**
 - 実行すると、全ての設定やログ、ユーザの作成したファイルが削除され、再起動
 - システムファイルは削除されない
 - **USB メモリや CF からの Format install**
 - **USB** メモリや **CF** に **Junos** イメージを書き込み、ブートローダーから **Junos** を再インストール
 - システムファイルを含むディスク上の全てのデータが削除され、新たに **Junos** がインストールされる
 - 実行方法は機種によって異なり、**JTAC** から指示された場合を除き、一般的に使用する必要はない

コントロールパケットのキャプチャ

以下のコマンドを使用することにより、コントロールパケット（**RE** が受信するパケット）をキャプチャする事が可能

```
root> monitor traffic interface xe-1/2/0.0
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on xe-1/2/0.0, capture size 96 bytes

11:39:06.772930 Out IP truncated-ip - 11 bytes missing! 192.168.1.1.bgp > 192.168.1.2.32794: P
635171747:635171766(19) ack 995070346 win 16384 <nop,nop,timestamp 3971359530 2610569>: BGP,
length: 19
11:39:06.803191 In IP 192.168.1.2.32794 > 192.168.1.1.bgp: . ack 19 win 5360 <nop,nop,timestamp
2637232 3971359530>
...
...
```

- このコマンドでキャプチャできるパケットは、**PFE** で処理されず **RE** で処理されるパケットに限られる
- **ICMP Echo（ping）** 等、**PFE** によってオフロード処理されるパケットは表示されないので注意
- パケット内容の詳細まで確認したい場合は **extensive** オプションなどを使用

groups / apply-groups

設定の一部を **group** という形で切り出し、**apply-groups** で任意の階層に適用する事が可能

- 例：全ての OSPF インターフェイスの **Hello-Interval** と **Dead-Interval** を変更

```
root# show groups
OSPF_COMMON {
  protocols {
    ospf {
      area <*> {
        interface <st*> {
          hello-interval 5;
          dead-interval 20;
        }
      }
    }
  }
}

root# show protocols ospf
apply-groups OSPF_COMMON;
area 0.0.0.0 {
  interface st0.1;
  interface st0.2;
  interface lo0.0 {
    passive;
  }
}
```

インターフェイス名やエリア名、IP アドレス等のユーザが自由入力する値は <*> とすると全てに適用される

特定のインターフェイスのみに適用したい場合などは、<st*> といったように一部の文字列を指定することも可能

自動的に共通設定が適用される



```
# show protocols ospf | display inheritance
area 0.0.0.0 {
  interface st0.1 {
    ##
    ## '5' was inherited from group 'OSPF_COMMON'
    ##
    hello-interval 5;
    ##
    ## '20' was inherited from group 'OSPF_COMMON'
    ##
    dead-interval 20;
  }
  interface st0.2 {
    ##
    ## '5' was inherited from group 'OSPF_COMMON'
    ##
    hello-interval 5;
    ##
    ## '20' was inherited from group 'OSPF_COMMON'
    ##
    dead-interval 20;
  }
  interface lo0.0 {
    passive;
  }
}
```

※ **Commit** しても **Config** はきちんとグループ化されたままとなる

実際に適用される設定を確認したい場合は、**show configuration | display inheritance** コマンドを使用

Prefix-list / apply-path

設定に含まれる IP アドレスから自動的にリストを生成し、**Firewall Filter** に適用することが可能

```
root# show protocols bgp
group GROUP-A {
    neighbor 1.1.1.1;
    neighbor 2.2.2.2;
}

root# show interfaces
ge-0/0/0 { unit 0 { family inet {
    address 1.1.1.0/30;
    } } }
ge-0/0/1 { unit 0 { family inet {
    address 2.2.2.0/30;
    } } }
fxp0 { unit 0 { family inet {
    address 192.168.1.10/24;
    } } }

root# show policy-options
prefix-list BGP-PEERS {
    apply-path "protocols bgp group <*> neighbor
<*>";
}
prefix-list LOCALNETS {
    apply-path "interfaces <ge-*> unit <*> family
inet address <*>";
}
```

IP アドレスが
自動的にコピーされる



```
root# show policy-options | display inheritance
prefix-list BGP-PEERS {
    ##
    ## apply-path was expanded to:
    ##     1.1.1.1/32;
    ##     2.2.2.2/32;
    ##
    apply-path "protocols bgp group <*> neighbor
<*>";
}
prefix-list LOCALNETS {
    ##
    ## apply-path was expanded to:
    ##     1.1.1.0/30;
    ##     2.2.2.0/30;
    ##
    apply-path "interfaces <ge-*> unit <*> family
inet address <*>";
}
```

※実際に適用される設定を確認したい場合は、**show configuration | display inheritance** コマンドを使用

オンライン・マニュアル

- 豊富な機能の **help** コマンド
 - **help topic** : プロトコルや機能の一般的な説明を表示
 - **help reference** : プロトコルや機能の設定方法を表示 (コマンド・レファレンス)
 - **help syslog** : **syslog** メッセージの説明

```
mike@juniper1> help topic interfaces address
Configuring the Interface Address
You assign an address to an interface by specifying the address when configuring the
protocol family. For the inet family, you configure the interface's IP address. For the
iso family, you configure one or more addresses for the loopback interface. For the ccc,
tcc, mpls, tnp, and vpls families, you never configure an address.b
```

Junos: help topic

コマンドの概要を確認することが可能

```
user@host> help topic ospf dead-interval
                Modifying the Router Dead Interval
```

If a router does not receive a hello packet from a neighbor within a fixed amount of time, the router modifies its topological database to indicate that the neighbor is nonoperational. The time that the router waits is called the router dead interval. By default, this interval is 40 seconds (four times the default hello interval).

To modify the router dead interval, include the dead-interval statement. This interval must be the same for all routers on a shared network.

dead-interval seconds;

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Junos: help reference

- コマンドのオンラインマニュアルを参照することが可能

```
user@host> help reference oam action
                                     action (OAM)

Syntax
action {
    syslog (OAM Action);
    link-down;
    send-critical-event;
}
Hierarchy Level
[edit protocols oam ethernet link-fault-management action-profile]
Release Information
Statement introduced in JUNOS Release 8.5.
...
Description
Define the action or actions to be taken when the OAM fault event occurs.
Usage Guidelines
See Specifying the Actions to Be Taken for Link-Fault Management Events.
```

Junos: help apropos

- 確実に覚えていないコマンド（うろ覚えの場合など）を文字列で検索することが可能

```
user@host# help apropos vstp
set logical-systems <name> protocols vstp
    VLAN Spanning Tree Protocol options
set logical-systems <name> protocols vstp disable
    Disable VSTP
set protocols vstp
    VLAN Spanning Tree Protocol options
set protocols vstp disable
    Disable VSTP
```

Configuration mode

```
user@host# > help apropos vstp
help topic stp vstp
    VLAN Spanning Tree Protocol instance configuration
help topic stp vstp-requirements
    Requirements, limitations for VLAN Spanning Tree Protocol
help reference stp vstp
    VLAN Spanning Tree Protocol configuration
help reference stp vlan-vstp
    VLAN configuration for VLAN Spanning Tree Protocol
```

Operation mode

CLI : trace / 充実した debug 機能

例 : OSPF Trace-option

注目したいパケットタイプを細かく指定することが可能

- Junos では, プロトコル別に **trace-options** を非常に細かく設定が可能
- この **trace** の出力先はファイル出力、あるいは **monitor** コマンドで **Real-time** に画面にてモニタ表示
- トラブルシューティングに役立つ情報を的確に抜き出すことが可能

```
lab@Router# set protocols ospf traceoptions flag ?
Possible completions:
  all                      Trace everything
  database-description    Trace database description packets
  error                   Trace errored packets
  event                   Trace OSPF state machine events
  flooding                Trace LSA flooding
  general                 Trace general events
  hello                   Trace hello packets
  lsa-ack                  Trace LSA acknowledgement packets
  lsa-request              Trace LSA request packets
  lsa-update              Trace LSA update packets
  normal                  Trace normal events
  packet-dump             Dump the contents of selected packet types
  packets                 Trace all OSPF packets
  policy                  Trace policy processing
  route                   Trace routing information
  spf                     Trace SPF calculations
  state                   Trace state transitions
  task                    Trace routing protocol task processing
  timer                   Trace routing protocol timer processing
```

CLI : monitor / リアルタイムにトラフィックを監視

- **monitor** コマンドで現在の I/F 別トラフィック状況を見ることが可能
- 表示は **AUTO** リフレッシュされるため、継続的なモニタリングが可能
- トラフィックの傾向や障害箇所の特定に役立ちます

```
10.0b2                               Seconds: 13                               Time: 14:50:48
Interface    Link  Input packets      (pps)      Output packets      (pps)
ge-0/0/0     Up    54175              (4)         4126                (0)
ge-0/0/1     Down  399                (0)         37                  (0)
ge-0/0/2     Up    5110               (1)         4224                (0)
ge-0/0/3     Down  0                  (0)         0                   (0)
ge-0/0/4     Down  0                  (0)         0                   (0)
ge-0/0/5     Down  0                  (0)         0                   (0)
ge-0/0/6     Down  0                  (0)         0                   (0)
```

```
Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D
```

rescue configuration

- 基本となる **Configuration** を予め定義（保存）することが可能

保存方法: > **request system configuration rescue save**

削除方法: > **request system configuration rescue delete**

- rescue configuration の反映方法
 - rollback コマンドからのロード
 - # rollback rescue

```
root# rollback rescue
load complete
root# commit
```

- ハードウェアからのロード

- SRX シリーズは **RESET CONFIG** ボタンを押すことでハードウェアからロードすることが可能
 - ※15秒以上押し続けると **factory default** がロードされる

例:
SRX300



- EX シリーズは **LCD** パネルでメンテナンスモードを操作することでハードウェアからロードすることが可能

例:
EX3300



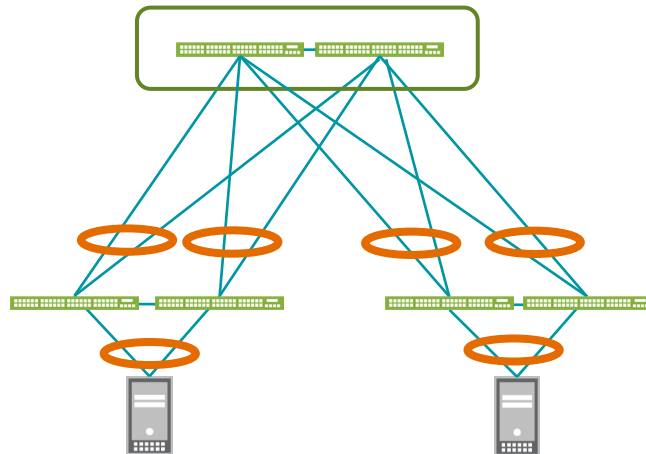


Appendix B: Multi-Chassis LAG

Juniper の提供する冗長化技術

筐体を跨いだ Link Aggregation (LAG) が組める技術として、主に以下のアーキテクチャを提供しています

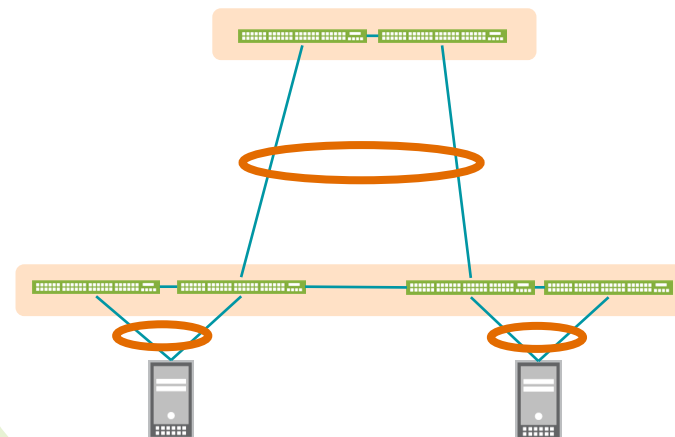
MC-LAG



スパンニングツリーなどを使用せずに標準化
プロトコルで L2 冗長を構成したい方

Virtual Chassis

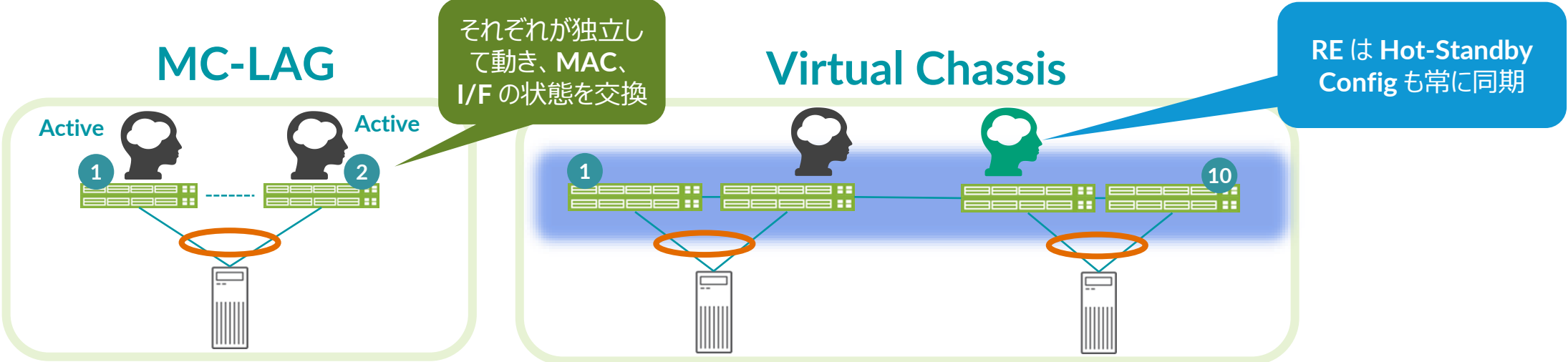
10 台までを、仮想的な 1 台の
シャーシー型スイッチとして管理



管理の負荷を下げつつ、柔軟なデザイン
を実現したい方

アーキテクチャ選択

MC-LAG vs Virtual Chassis



	MC-LAG	Virtual Chassis
コントロールプレーン	Active-Active	Active-Standby
データプレーン	Active-Active	Active-Active
管理	2 台別々	10 台まで 1 台として管理
設定同期	手動 (※Roadmap)	自動
対向デバイスから見た L2 ネイバー	1 台に見える	1 台に見える
対向デバイスから見た L3 ネイバー	2 台に見える	1 台に見える
バージョンアップ	1 台ずつ (ISSU)	NSSU/ISSU (※Roadmap)

スイッチ台数が増え
てくると、管理面で
差が出てきます

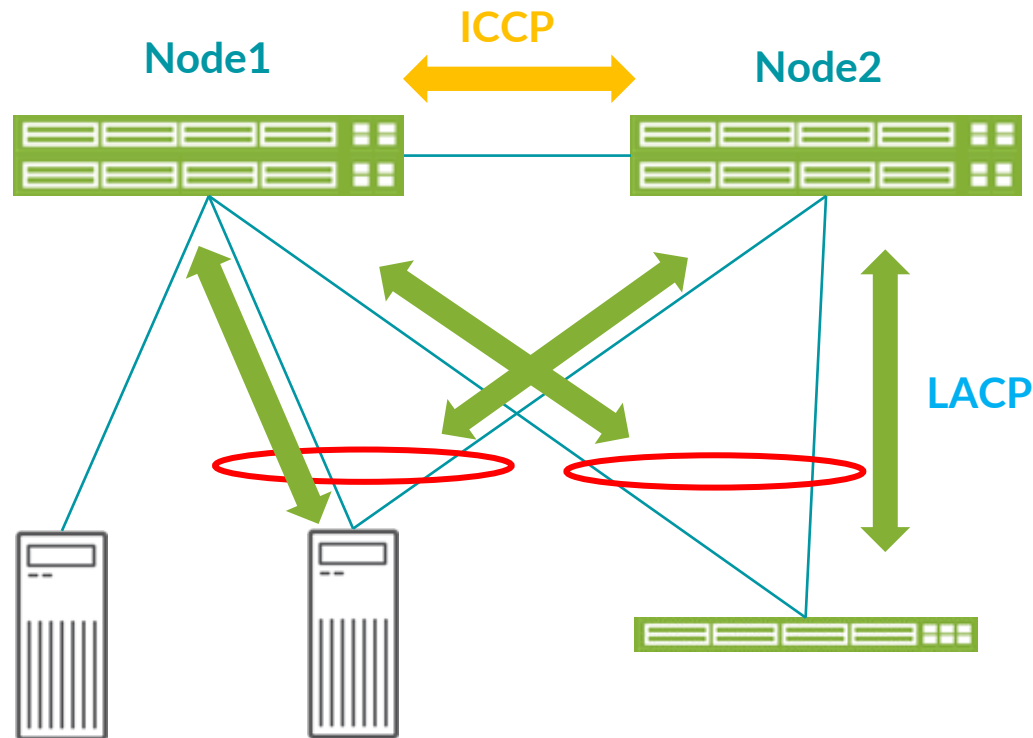
MC-LAG 対応プラットフォーム

	MX シリーズ	QFX シリーズ	EX9200	EX4600 シリーズ	EX4300 シリーズ	その他の EX シリーズ
MC-LAG	✓	✓	✓	✓	✓	
Active / Active 構成	✓	✓	✓	✓	✓	
Active / Standby 構成	✓		✓			
VRRP との組合せ	✓	✓	✓	✓	✓	
L2VPN (MPLS) との組合せ	✓	✓	✓			
VPLS との組合せ	✓					

MC-LAG 基本構成

- MC-LAG を構成する上での基本

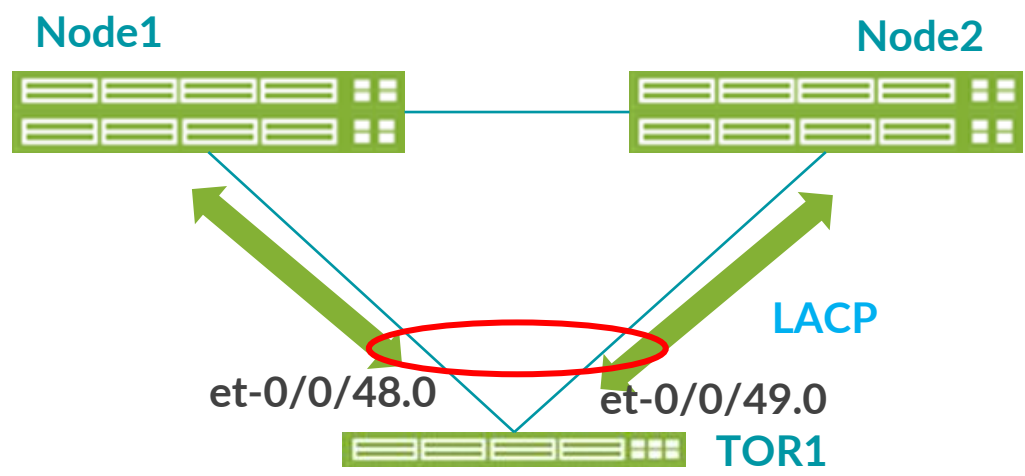
MC-LAG を構成するスイッチはどちらも **Active**（Master / Backup 等の関係では無い）のため、「Node1」、「Node2」と呼ばれる



Node1・2 の間は **MAC** アドレスや **Link** のステータスを同期 (**ICCP**)

MC-LAG につながる **LAG** 機器とは **LACP** でステータスを交換

MC-LAG 基本構成



スイッチ TOR1 から見ると MC-LAG は単なる LAG にしか見えない

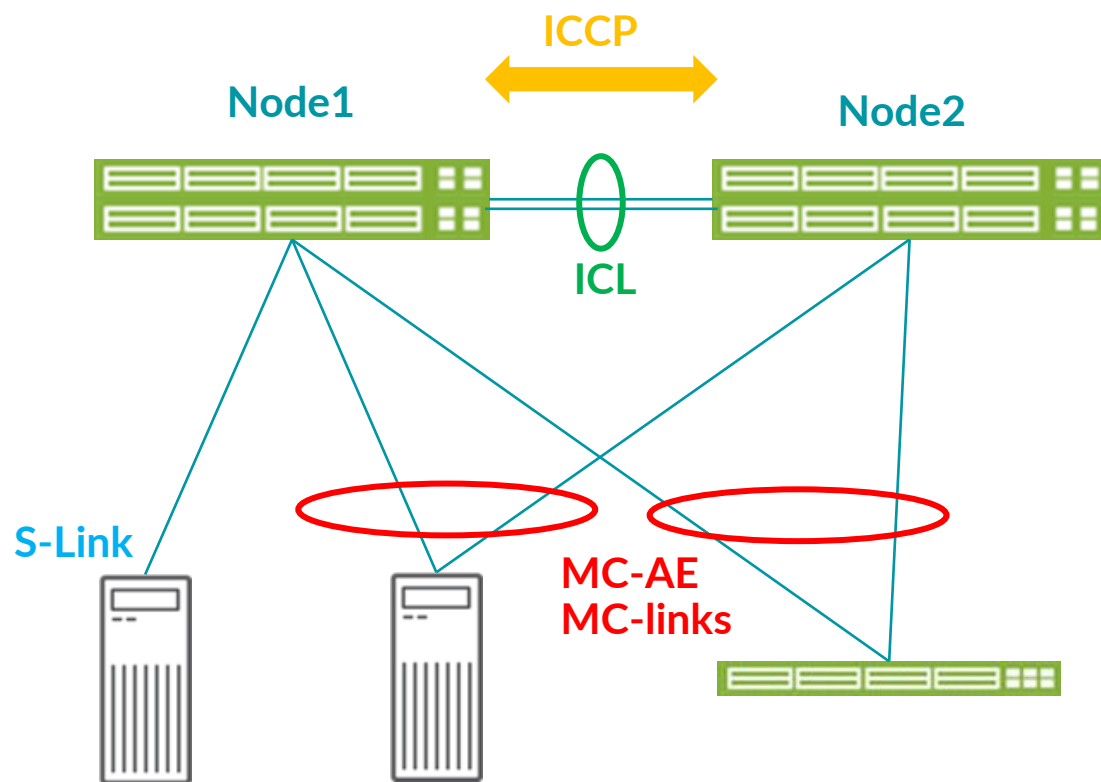
LACP で見ても、どちらの MAC も同じ MAC が見える

TOR1 での LACP のステータス出力例

LACP info:	Role	System priority	System identifier	Port priority	Port number	Port key
et-0/0/48.0	Actor	127	54:1e:56:69:4e:00	127	1	3
et-0/0/48.0	Partner (Node1)	127	00:00:ae:00:00:02	127	2	1002
et-0/0/49.0	Actor	127	54:1e:56:69:4e:00	127	2	3
et-0/0/49.0	Partner (Node2)	127	00:00:ae:00:00:02	127	32770	1002

用語の整理

- MC-LAG は各ベンダーで用語が異なりますが、ジュニパーでは以下の用語を使用



ICCP (Inter-chassis control protocol) :
MAC や Link の状態を Node 間で共有する為の制御通信用途で、TCP セッションにより確立される

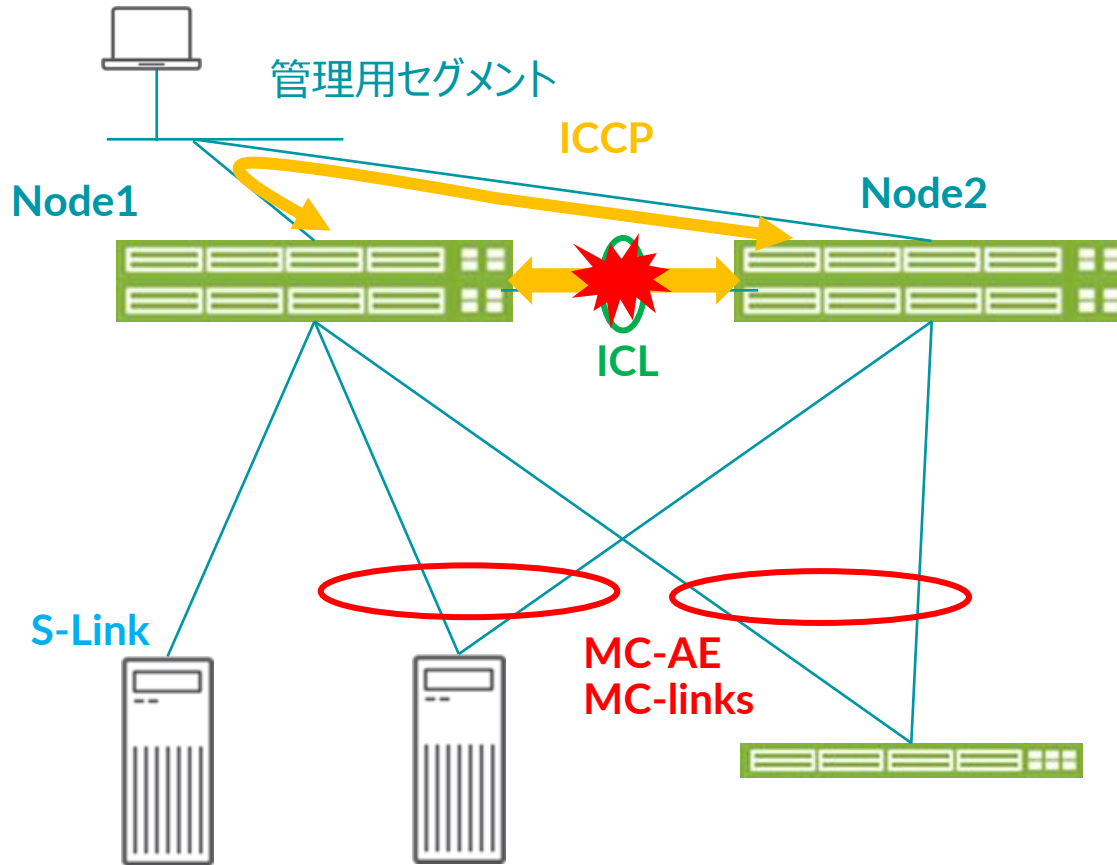
ICL (Inter-Chassis Link) :
スイッチ間の物理 Link (ICL-PL と呼ばれる)
出来る限りここを LAG で構成するデザインが推奨

MC-AE (または MC-links) :
スイッチまたぎの LAG を指す
AE は Aggregated Ethernet の略

S-LINK (Single-homed Link) :
冗長されていない Link
既存の収容、NW 移行やメンテナンスなどにより、一時的にこの構成になりえる

用語の整理(つづき)

- 出来る限り、**ICCP** の接続用途で管理セグメントも使用することが推奨



ICCP が切れてしまう状況は、**Split Brain** と呼ばれ絶対に避けるべき状況

ICL のバックアップとして、管理セグメントを使った **ICCP** のやりとりが可能 (backup-liveness-detection)

ただし、ユーザパケットは転送されない
あくまで **Split Brain** の状態を避ける為の最終手段

※ ICL 故障発生時の動作の詳細は以下で確認できます。
https://www.juniper.net/documentation/en_US/junos/topics/concept/mc-lag-feature-concepts.html

L3 の冗長構成は？

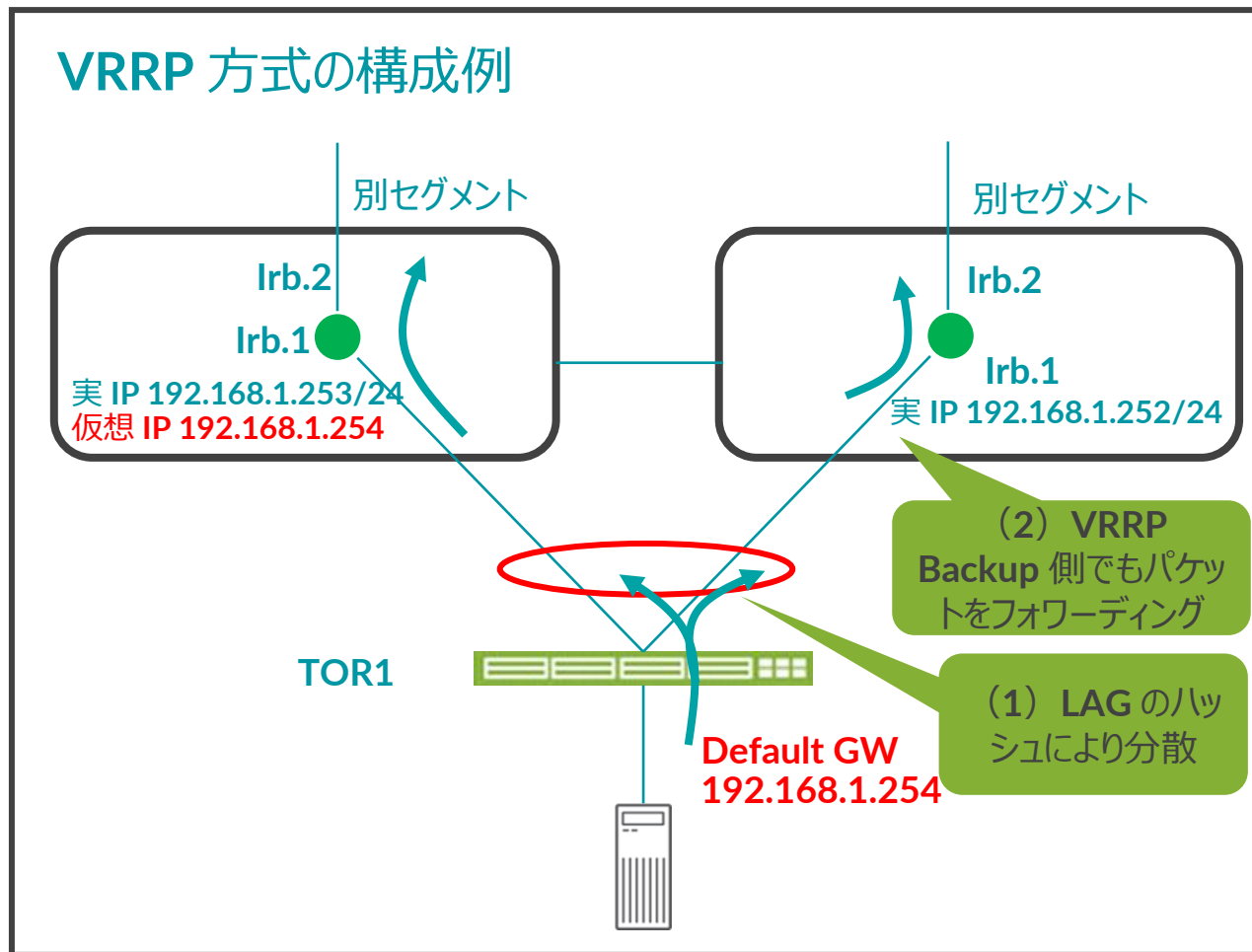
デフォルト GW の冗長について

2つの方式が存在

- VRRP over IRB 方式
 - Node 同士で VRRP を構成
- MAC Sync 方式
 - Node 同士で同じ IP、MAC を構成

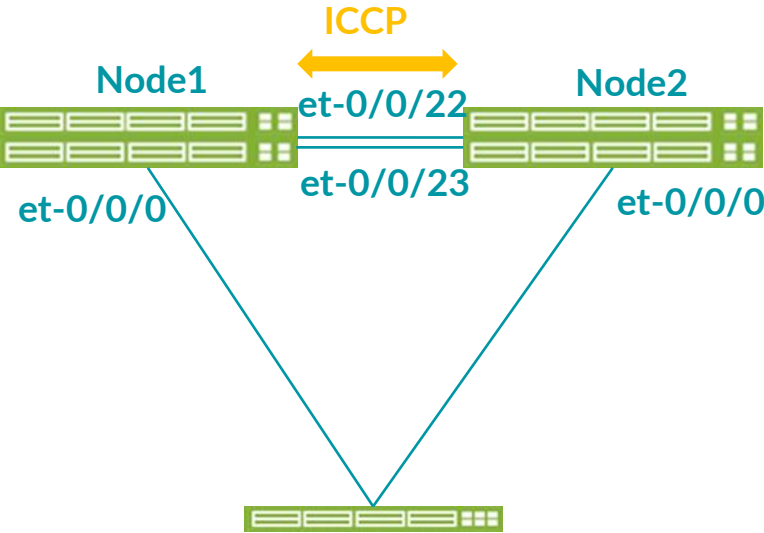
※ 以下の理由から **VRRP 方式を推奨**

1. TOR1 からみて Uplink は LAG のため、トラフィックは分散可能
2. VRRP Backup 側でも受け取ったユーザトラフィックは転送できる実装の為、ICL を通ったり、Uplink が偏ったりしない
3. MAC Sync 方式では、Routing Protocol が話せない（あくまで Node 間で同期しているのは MC-LAG 関連情報のみ）



設定方法：

- 基礎となる設定



設定項目	Node1	Node2	備考
Device-count	10	10	必要な MC-LAG 数 +1 を設定
switch-options service-id	16384	16384	2 台とも同じ値にする
ICCP 用 I/F	irb.4000	irb.4000	irb + unit 番号 (Vlan-id と同じが推奨)
ICCP 用 Vlan	4000	4000	渡りの LAG にこの Vlan を 所属させる
ICCP 用 IP アドレス	192.168.254.26/24	192.168.254.27/24	ICCP だけで利用のため、 /30 なども可

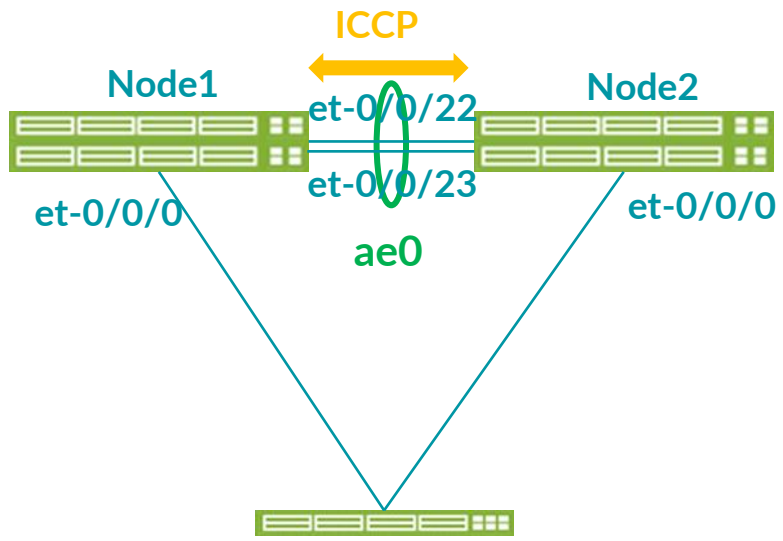
※ Node1 の設定例

```
set chassis aggregated-devices ethernet device-count 10
set switch-options service-id 16384
set interfaces irb unit 4000 family inet address 192.168.254.26/24
set vlans VLAN4000 vlan-id 4000
set vlans VLAN4000 l3-interface irb.4000
```

例の IRB アドレス値を変更することによって、Node2 用の設定となる
どちらか一方にだけ投入する設定ではない

設定方法：

- 基礎設定その 2



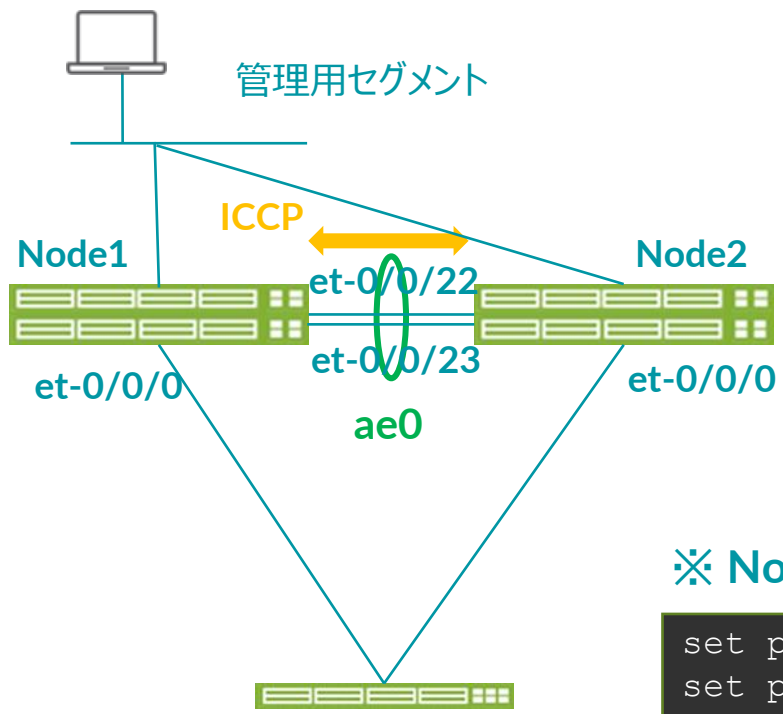
設定項目	Node1	Node2	備考
ICCP 用 IP アドレス	192.168.254.26/24	192.168.254.27/24	VLAN4000 だけで利用のため、/30 でも可
ICL Interface の ID	ae0	ae0	ae は aggregated-ethernet の略、LAG 用仮想 I/F 名
ae0 に所属させる物理 I/F	et-0/0/22 et-0/0/23	et-0/0/22 et-0/0/23	
その他	LACP Fast モード Vlan4000	LACP Fast モード Vlan4000	LACP と Vlan4000 を ae0 に設定

※ Node1 の設定例

```
set multi-chassis multi-chassis-protection 192.168.254.27 interface ae0
set interfaces et-0/0/22 ether-options 802.3ad ae0
set interfaces et-0/0/23 ether-options 802.3ad ae0
set interfaces ae0 aggregated-ether-options lacp active periodic fast
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members VLAN4000
```

設定方法：

- 基礎設定その 3



設定項目	Node1	Node2	備考
ICCP 用 IP アドレス	192.168.254.26/24	192.168.254.27/24	VLAN 4000 だけで利用のため、/30 でも可
session-establishment-hold-time	100	100	ICCP セッション確立までの時間(秒)
BFD minimum-interval	1000	1000	お互いの ICCP 間で行う BFD 死活監視の間隔 (msec) 1000 以上 に設定
BFD multiplier	3	3	回数 minimum-interval x multiplier = ダウンまでの時間
backup-liveness-detection	172.27.113.26	172.27.113.27	管理 I/F に付与した IP アドレスを指定

※ Node1 の設定例

```
set protocols iccp local-ip-addr 192.168.254.26
set protocols iccp peer 192.168.254.27 session-establishment-hold-time 100
set protocols iccp peer 192.168.254.27 liveness-detection minimum-interval 1000
set protocols iccp peer 192.168.254.27 liveness-detection multiplier 3
set protocols iccp peer 192.168.254.27 backup-liveness-detection backup-peer-ip 172.27.113.27
```

設定方法：

- 基本設定の確認

- 正しく接続が行われた場合の表示状態：

```
lab@node1# run show iccp
Redundancy Group Information for peer 192.168.254.27
  TCP Connection      : Established
  Liveliness Detection : Up

Backup liveness peer status: Up

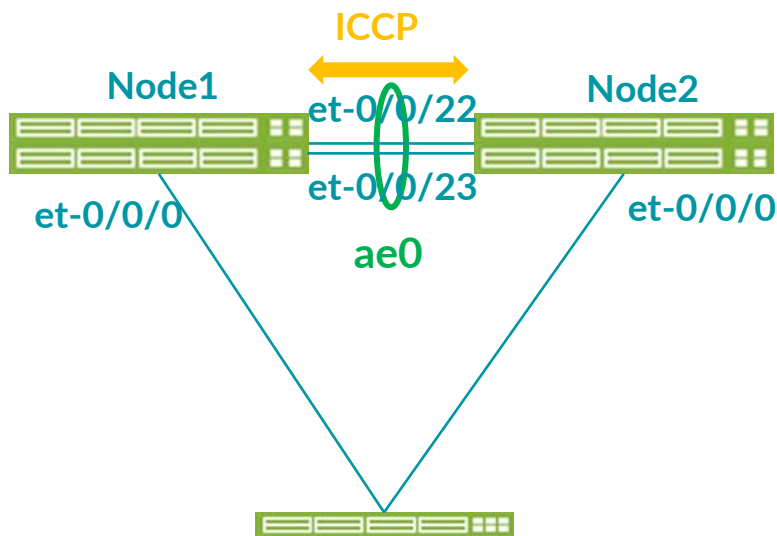
Client Application: MCSNOOPD
Client Application: l2ald_iccpd_client
Client Application: lacpd
```

- ae0 が Up にならない場合：

- ▶ 最初に使用する LAG 数の登録が必要

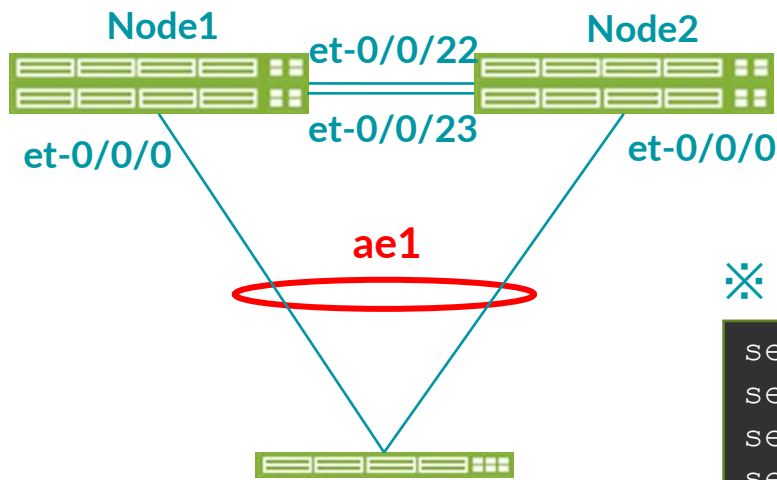
```
set chassis aggregated-devices ethernet device-count 10
```

※「10」で設定の場合 ae0～ae9 までの I/F が作成される



設定方法：

- MC-Links の設定
- 次に TOR スイッチを収容する LAG を設定します。



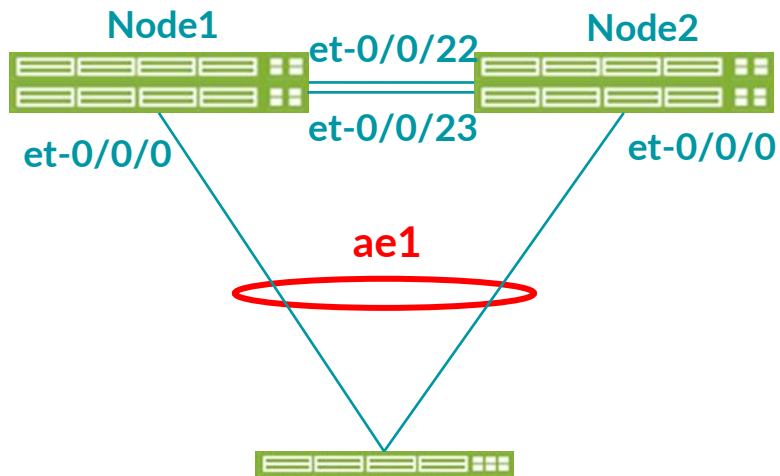
設定項目	Node1	Node2	備考
LAG I/F 名	ae1	ae1	
LACP system-id	00:00:ae:00:00:01	00:00:ae:00:00:01	同じ値を設定、LAG 毎に変更
LACP admin-key	1001	1001	同じ値を設定、LAG 毎に変更
mc-ae mc-ae-id	1001	1001	同じ値を設定、LAG 毎に変更
mc-ae chassis-id	0	1	Node 毎に変更
mc-ae status-control	Active	Standby	
mc-ae init-delay-time	60	60	I/F が Up となってから LACP が distributing となるまでの時間 電源投入時など、Protocol が Up となるまでの時間を待たせることが可能

※ Node1 の設定例

```
set interfaces et-0/0/0 ether-options 802.3ad ae1
set interfaces ae1 aggregated-ether-options lacp active periodic fast
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:ae:00:00:01
set interfaces ae1 aggregated-ether-options lacp admin-key 1001
set interfaces ae1 aggregated-ether-options mc-ae mc-ae-id 1001
set interfaces ae1 aggregated-ether-options mc-ae chassis-id 0
set interfaces ae1 aggregated-ether-options mc-ae mode active-active
set interfaces ae1 aggregated-ether-options mc-ae status-control active
set interfaces ae1 aggregated-ether-options mc-ae init-delay-time 60
```

設定方法：

- MC-Links の確認

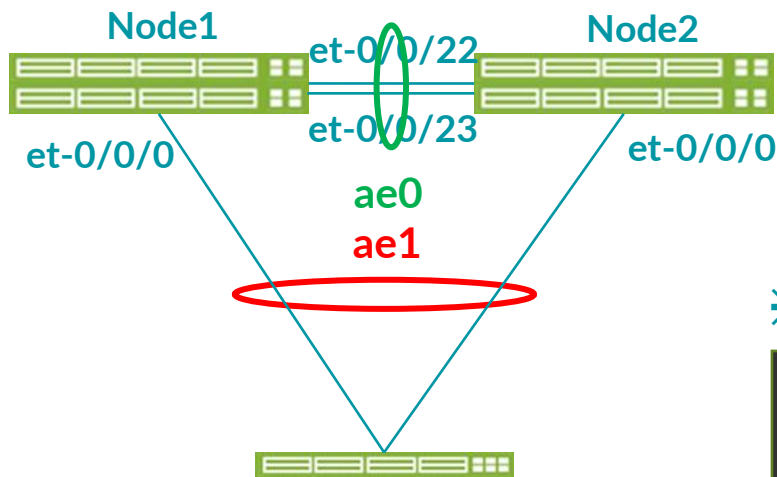


- 対応の設定に問題がなく、正しく接続が行われた場合の表示状態：

```
lab@node1# run show interfaces mc-ae id 1001
Member Link                : ae1
Current State Machine's State: mcae active state
Local Status                : active
Local State                 : up
Peer Status                 : active
Peer State                  : up
  Logical Interface         : ae1.0
  Topology Type              : bridge
  Local State                : up
  Peer State                 : up
  Peer Ip/MCP/State         : 192.168.254.27 ae0.0 up
```

設定方法：

- MC-Links への VLAN の組み込み
- VLAN1001 を ae1 に追加 ※ ICL (ae0) にも追加するのを忘れずに



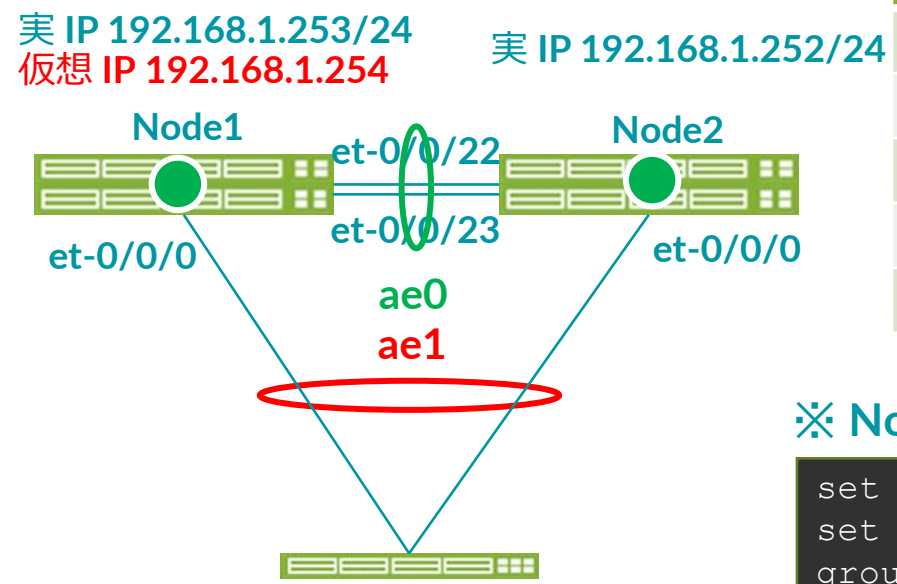
設定項目	Node1	Node2	備考
vlan 名	V1001	V1001	
vlan-id	1001	1001	

※ Node1 の設定例

```
set vlans v1001 vlan-id 1001
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members v1001
set interfaces ae1 unit 0 family ethernet-switching storm-control default
set interfaces ae0 unit 0 family ethernet-switching vlan members v1001
```


設定方法：

- [Option] L3 Routing（Default Gateway）の設定
- VLAN1001 にサーバのデフォルトゲートウェイとなるアドレスを設定



設定項目	Node1	Node2	備考
I/F 名	irb unit 1001	irb unit 1001	
実 IP	192.168.1.253/24	192.168.1.252/24	
仮想 IP	192.168.1.254	192.168.1.254	
Priority	200	100	
Accept-data	設定する	設定する	

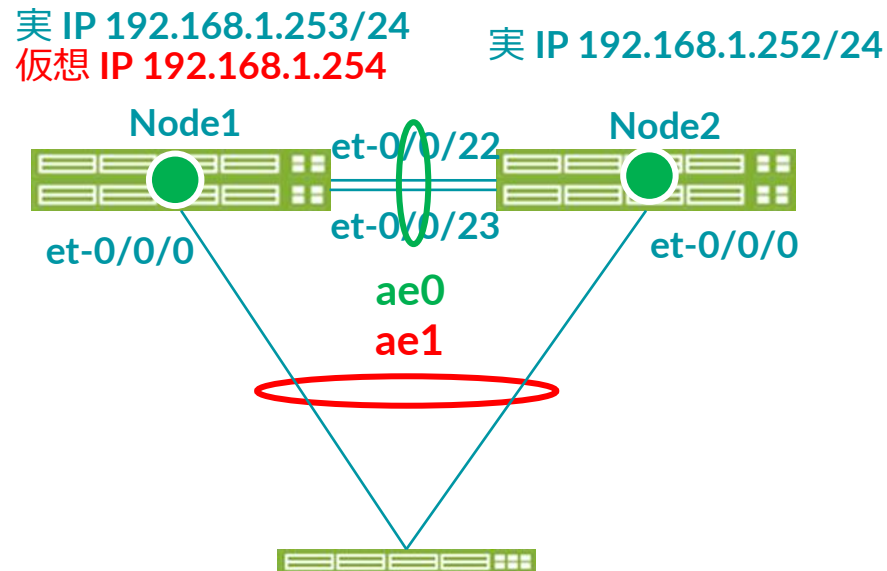
※ Node1 の設定例

```
set vlans v1001 13-interface irb.1001
set interfaces irb unit 1001 family inet address 192.168.1.253/24 vrrp-
group 1 virtual-address 192.168.1.254
set interfaces irb unit 1001 family inet address 192.168.1.253/24 vrrp-
group 1 priority 100
set interfaces irb unit 1001 family inet address 192.168.1.253/24 vrrp-
group 1 accept-data
```

設定方法：

L3 Routing（Default Gateway）の確認

- Default Gateway アドレスが冗長されているかを確認
- 下記の状態表示が確認されない場合：
 - ICL（ae0）に VLAN1001 が設定されているかを確認



Node1

```
{master:0}[edit]
lab@node1# run show vrrp
Interface      State      Group  VR state VR Mode  Timer  Type  Address
irb.1001       up         1      master  Active   A    0.359 lcl   192.168.1.253
                                     vip     192.168.1.254
```

Node2

```
lab@node2# run show vrrp
Interface      State      Group  VR state VR Mode  Timer  Type  Address
irb.1001       up         1      backup  Active   D    2.718 lcl   192.168.1.252
                                     vip     192.168.1.254
                                     mas     192.168.1.253
```



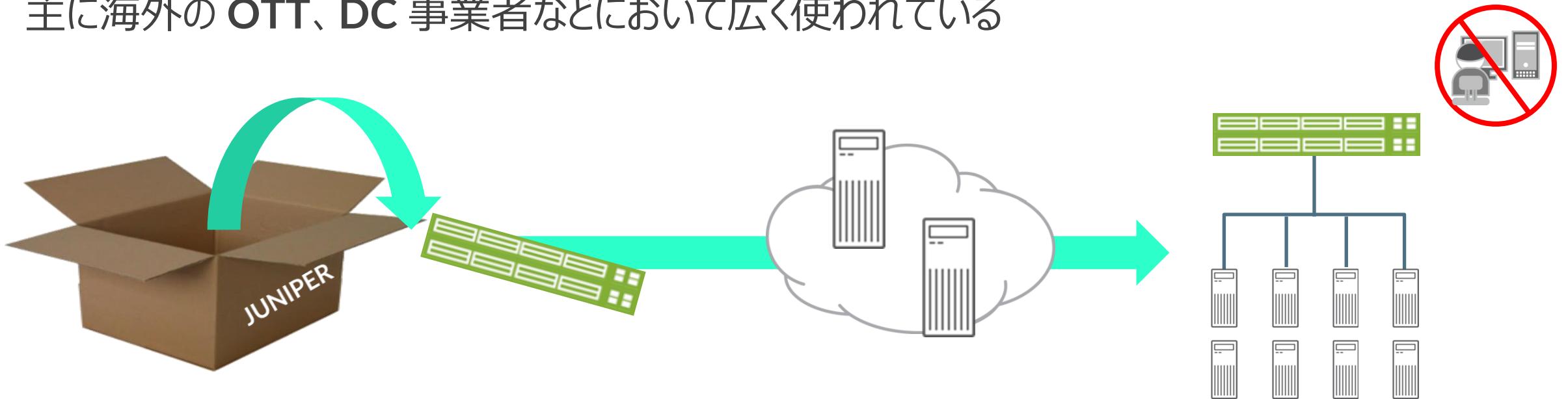
Appendix C: Zero Touch Provisioning

ZTP (Zero Touch Provisioning) とは

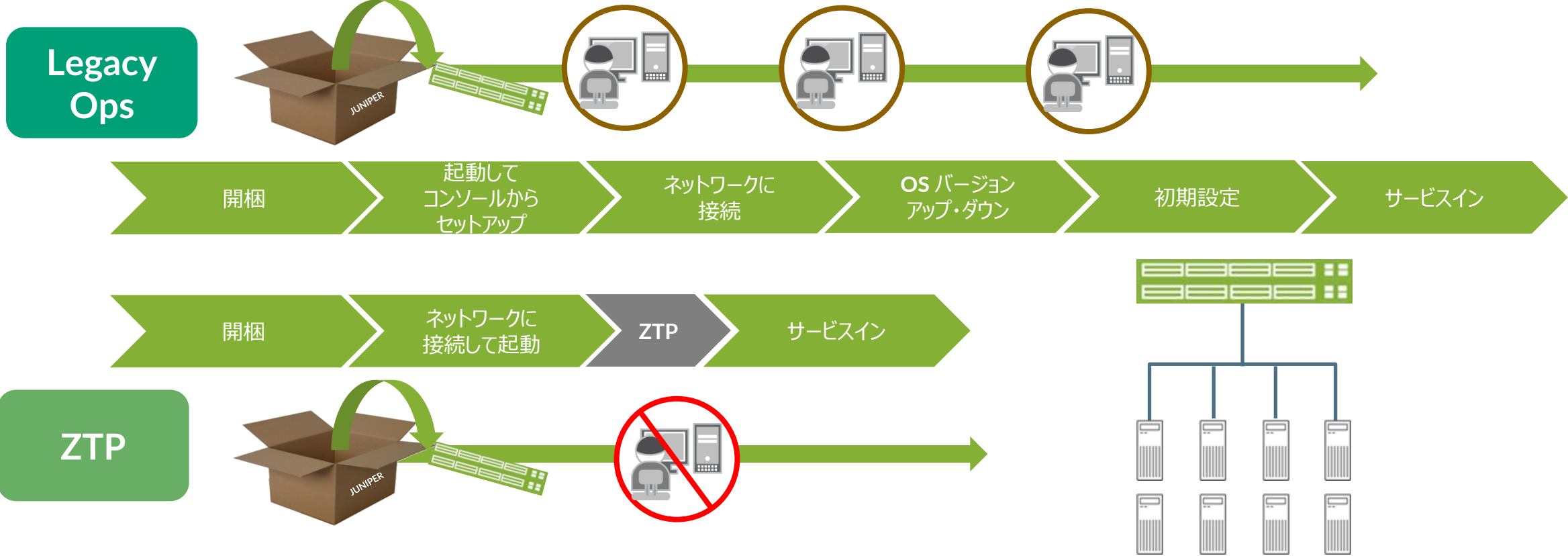
DC に ToR スイッチを新設するオペレーションは簡単だが、大量のスイッチを展開することを考えると手間は膨大

ZTP とは、スイッチの初期導入時において **Junos** のバージョンとコンフィグを自動でプロビジョニングする機能（Junos 12.2 よりサポート）

主に海外の **OTT**、**DC** 事業者などにおいて広く使われている



ZTP と従来のオペレーションとの比較



膨大な数の **ToR** を設置しなくてはならない **DC** 等の場合には、
ZTP による **OPEX** の削減効果は絶大！

Junos ZTP Overview – Components

自動的なプロビジョニングを前提とした DC 向けイーサネット・スイッチの実装

- 自動的な OS アップグレード – 管理者の手をわずらわせない実装
- 自動的なベースコンフィグレーションの投入 – 管理者の手をわずらわせない実装
- Junos 12.2 よりすべての Juniper スイッチ（EX、QFX、OCX）でサポート

工場出荷状態のスイッチ

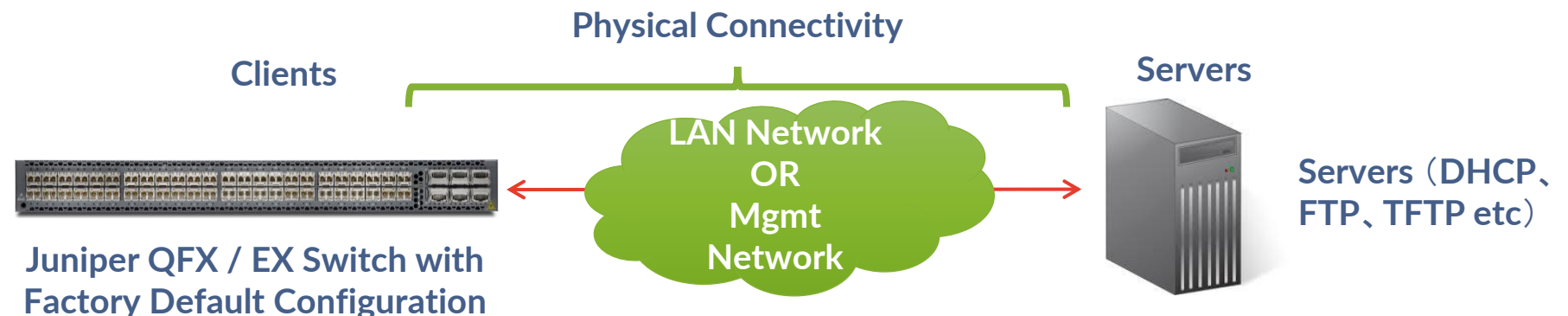
- Zeroized configuration - <request system zeroize>
- 管理ネットワークに接続して電源を投入するだけ

DHCP Server:

- 自動プロビジョニングの動作を指定
- OS イメージと設定ファイルの配備場所を指定

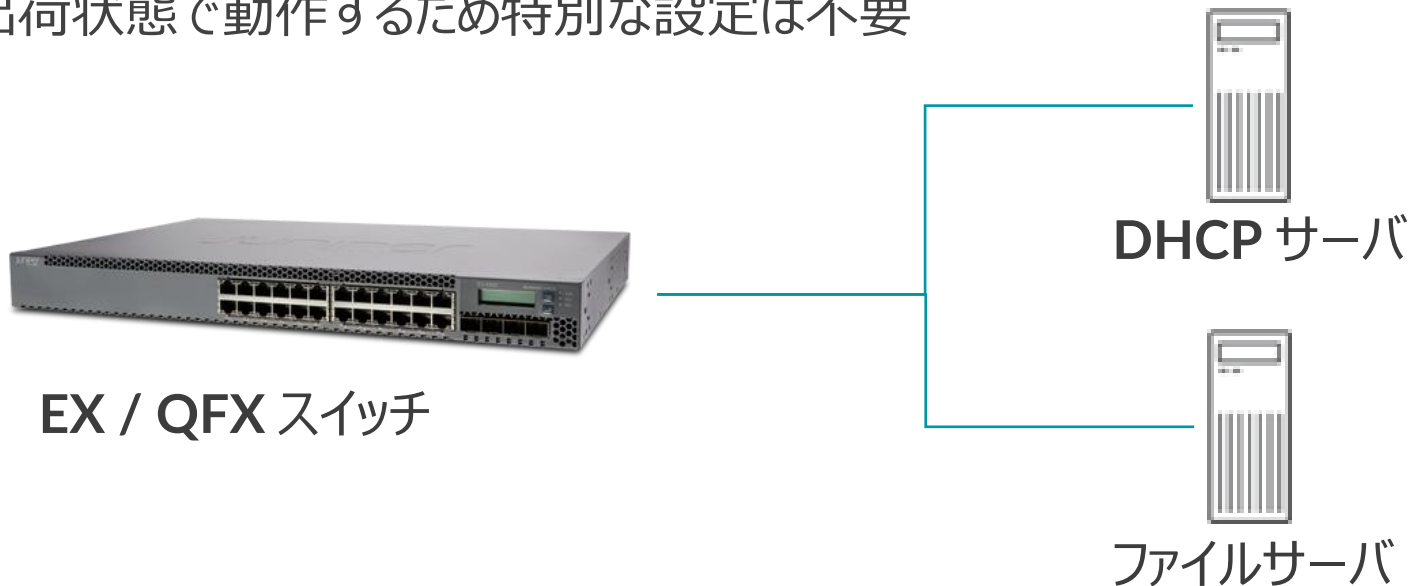
HTTP / FTP / TFTP Server

- OS イメージと設定ファイルの格納場所



ZTP のコンポーネント

- ゼロタッチ：装置にコンソールなどでのログインが不要
 - 電源を入れるだけ！
- 用意するのは **DHCP** サーバとファイルサーバの 2 つ
- ネットワーク経由で自動的に **OS** や設定情報を装置に転送し反映
- 工場出荷状態で動作するため特別な設定は不要



- 装置に IP アドレスを付与
- TFTP サーバのアドレスを通知
- 取得すべき Config ファイル名を通知

- Config ファイル
- OS ファイル

動作シーケンス

- スイッチはシリアルと **MAC** アドレスを含む **DHCP** リクエストを送信
- **DHCP Option** で **TFTP** サーバの **IP** アドレスと **OS / Config** のファイル名を通知

1. デフォルト設定で起動
ZTP スタート！



EX / QFX スイッチ

6. ダウンロードしたファイルで
OS と **Config** を書き換え
commit

2. DHCP Discover / Request

3. DHCP Offer / ACK
(ファイルサーバ + ファイル名)

4. File Request
(指定されたファイル名)

5. Download files



- 装置に **IP** アドレスを付与
- **TFTP** サーバのアドレスを通知
- 取得すべき **Config** ファイル名を通知

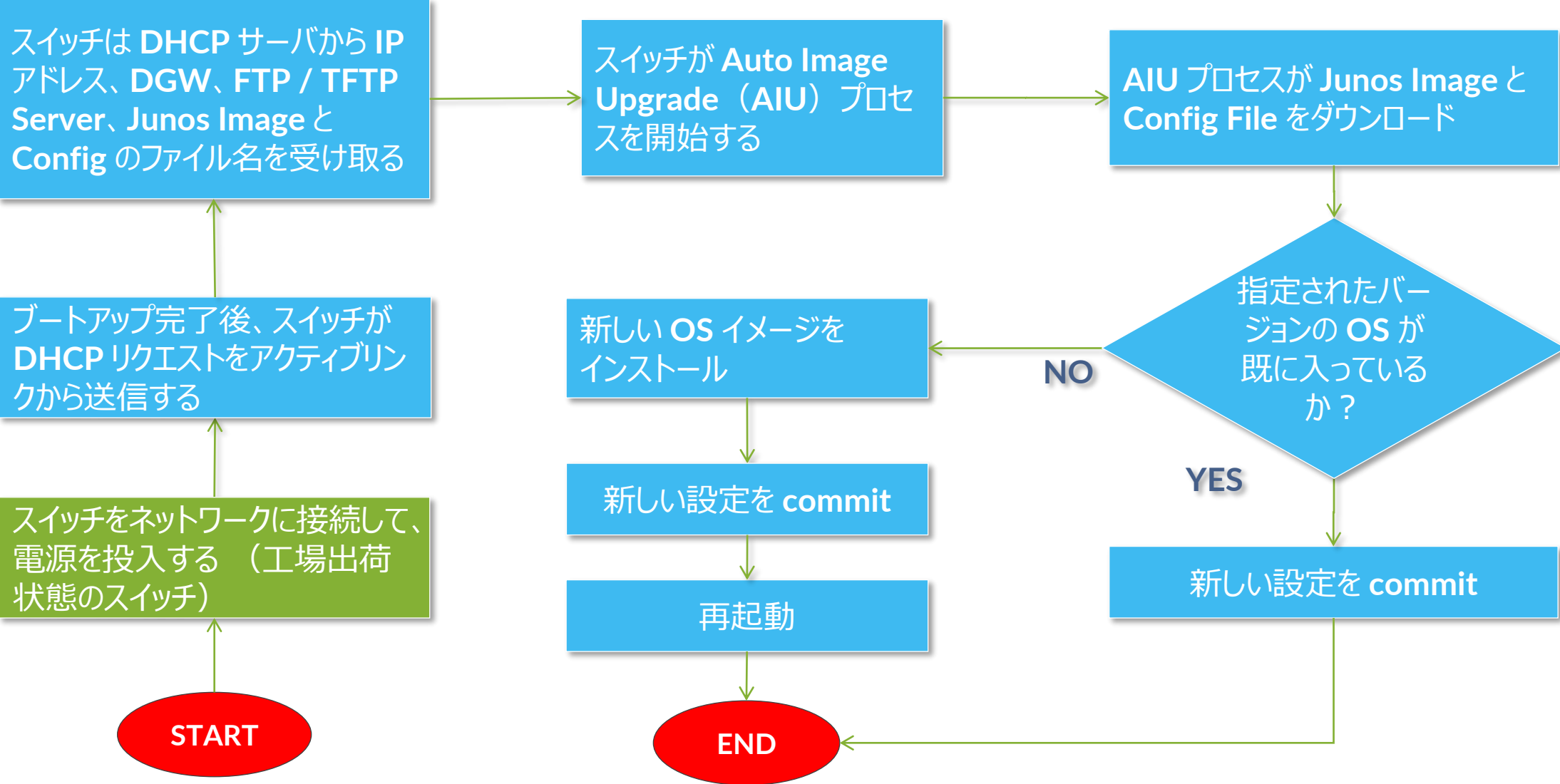
DHCP サーバ



- **Config** ファイル
- **OS** ファイル

ファイルサーバ
(**TFTP** / **FTP** / **HTTP**)

Junos ZTP の流れ



筐体の識別

DHCP サーバの設定

```
ddns-update-style none;
option option-66 code 66 = string;
option space NEW_OP;
option NEW_OP.config-file-name code 1 = text;
option NEW_OP-encapsulation code 43 = encapsulate NEW_OP;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.1 192.168.1.200;
    default-lease-time 6000;
    max-lease-time 7200;
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;

    host switch1 {
        hardware ethernet 2c:6b:f5:3a:6e:41;
        fixed-address 192.168.1.11;
        option NEW_OP.config-file-name "switch1.cfg";
        option option-66 "192.168.1.100";
    }

    host switch2 {
        hardware ethernet 64:87:88:B7:45:81;
        fixed-address 192.168.1.12;
        option NEW_OP.config-file-name "switch2.cfg";
        option option-66 "192.168.1.100";
    }
}
```

筐体背面又は底面のシールに筐体の **MAC** が記載



筐体の MAC アドレスに +1 したものを設定に記述

JUNOS ZTP – DHCP Server Configuration (サンプル)

```
option space NEW_OP;
option NEW_OP.image-file-name code 0 = text;
option NEW_OP.config-file-name code 1 = text;
option NEW_OP-encapsulation code 43 = encapsulate NEW_OP;

group {
    option tftp-server-name "17.176.31.71";
    option log-servers 17.176.31.72;
    option ntp-servers 17.176.31.73;
    option NEW_OP.image-file-name "/images/jinstall-qfx.tgz";
    option NEW_OP.transfer-mode "ftp";
    host tor-qfx5100-1 {
        hardware ethernet 88:e0:f3:71:a0:82;
        fixed-address 172.16.31.19;
        option host-name "tor-qfx5100-1";
        option NEW_OP.config-file-name "tor-qfx5100-1.config";
    }
    host tor-qfx5100-2 {
        hardware ethernet f8:c0:01:c6:96:81;
        fixed-address 172.16.31.20;
        option host-name "tor-qfx5100-2";
        option NEW_OP.config-file-name "tor-qfx5100-2.config";
    }
}
```

Vendor Specific Options
(Auto Image Upgrade に必要)

Syslog and NTP servers

投入される OS イメージ
シンボリックリンクを指定することも
可能

Auto Configuration で指定され
る設定ファイル

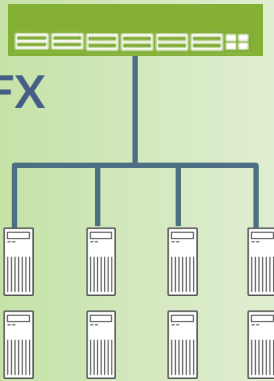
MAC アドレスから IP アドレスお
よびシステムのホストネームへの
マッピング

ZTP が提供するもの



効率

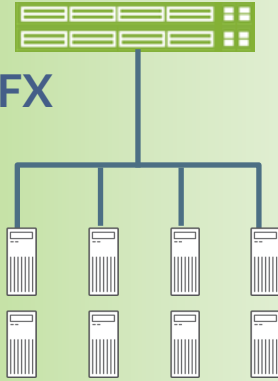
Any EX & QFX
Switches



導入にかかる
時間を削減

一貫性

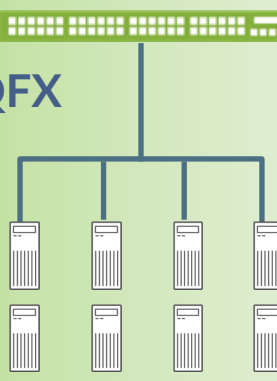
Any EX & QFX
Switches



設定ミスの削減

可用性

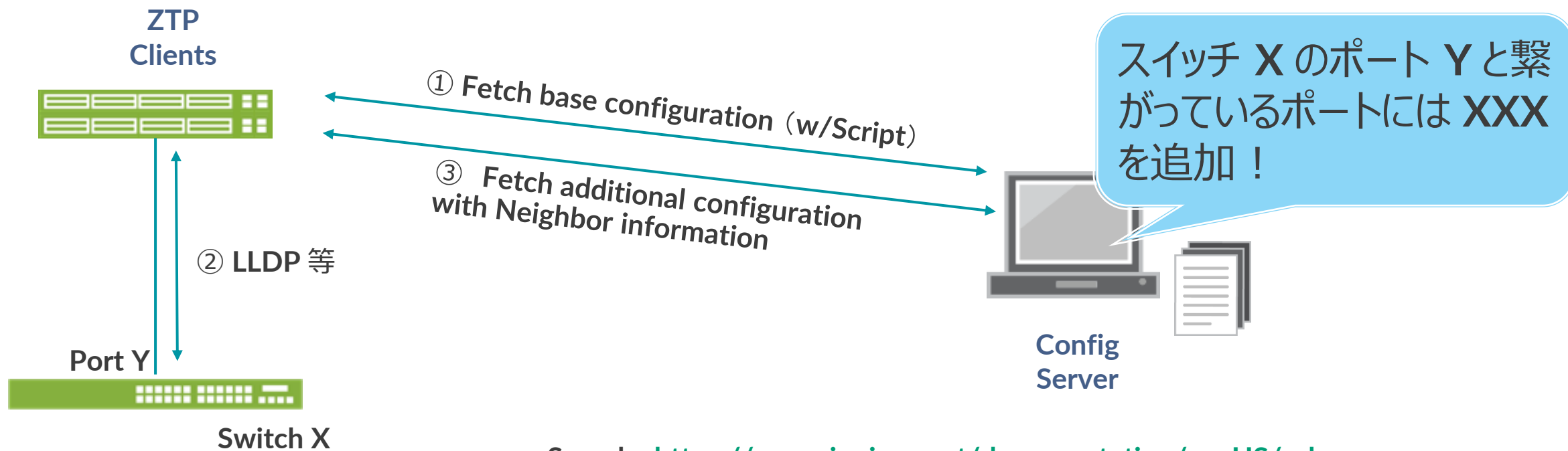
Any EX & QFX
Switches



機器故障時の交換に
かかる時間を削減

ZTP + Script

MAC アドレス / Serial ベースではなく、ネットワークの情報を元に **Config** を投入



Sample: https://www.juniper.net/documentation/en_US/release-independent/ncs/topics/example/ncs151-example-config-ztp.html



THANK YOU

JUNIPER
NETWORKS | Driven by
Experience™