



JUNOS Hands On Training “SRX” 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. 30
Junos CLI 操作 ～ Operational モード～	P. 37
Junos CLI 操作 ～ Configuration モード～	P. 56
Junos システム設定	P. 73
Junos インタフェース設定	P. 81
Junos 経路設定	P. 89

トレーニング概要「サービスゲートウェイ “SRX” コース」

トレーニング内容（後半）	記載ページ
Juniper SRX シリーズ製品紹介	P. 95
LAB.1 Junos の基本的な操作・設定	P. 101
LAB.2 Firewall の設定	P. 114
LAB.3 NAT の設定	P. 135
LAB.4 Chassis Cluster の設定	P. 155
Appendix	P. 192



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

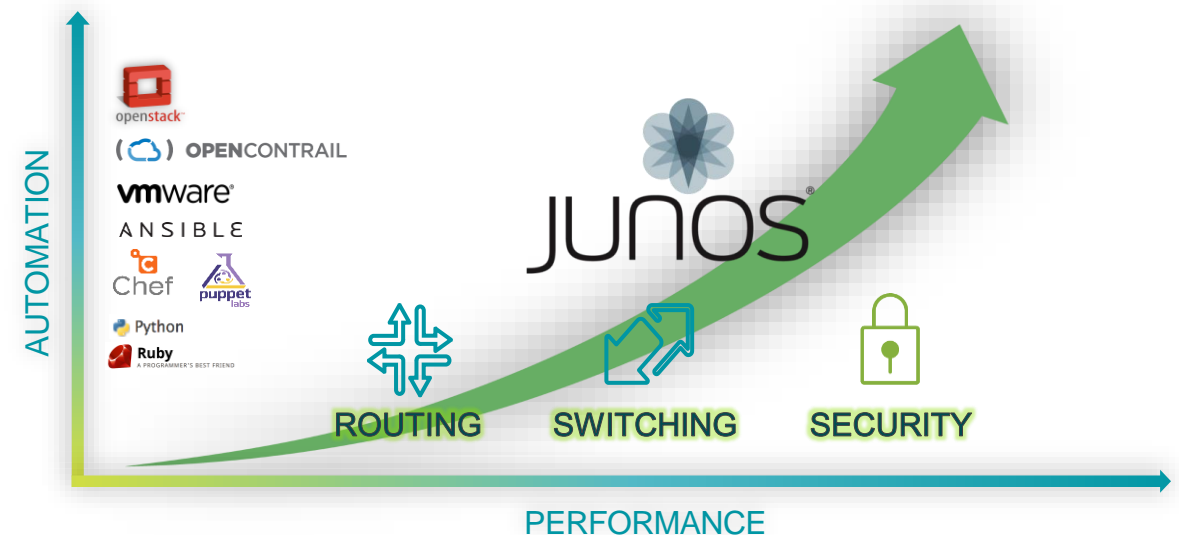
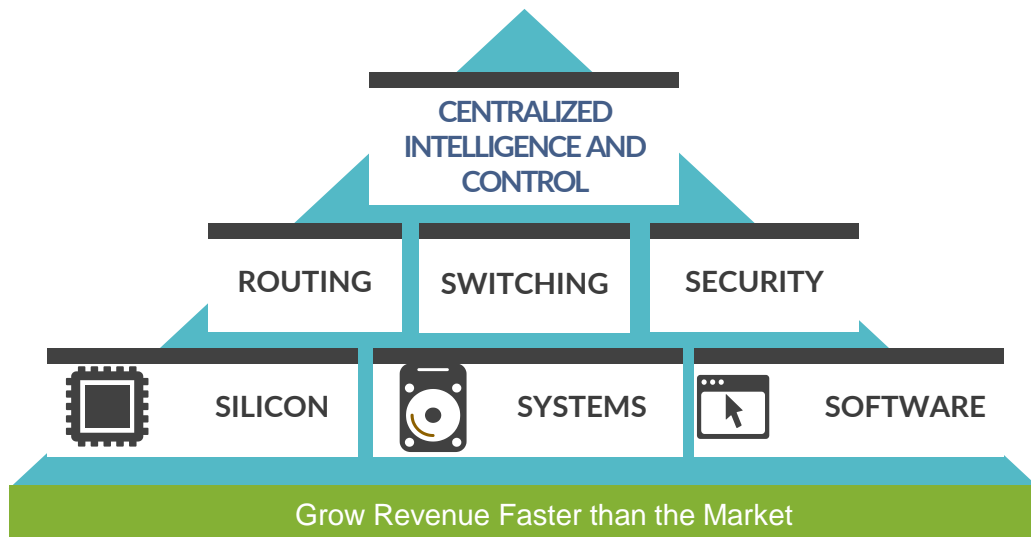
Go-To-Market : ハイパフォーマンスネットワーキングをビジネスの基盤と位置付けるお客様とパートナー様に価値を提供

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



SSG Series

SSG550M
SSG520M
SSG350M
SSG320M
SSG140



ISG Series

ISG2000
ISG1000



JUNOS: THE POWER OF ONE INTEGRATED ARCHITECTURE

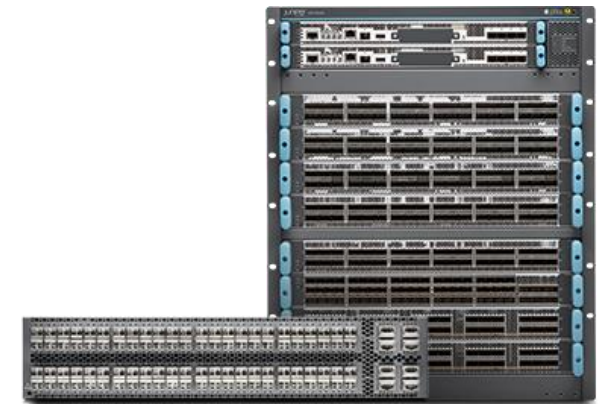
Datacenter Service Gateway
SRX series



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
IOS-NX
IOS-XR

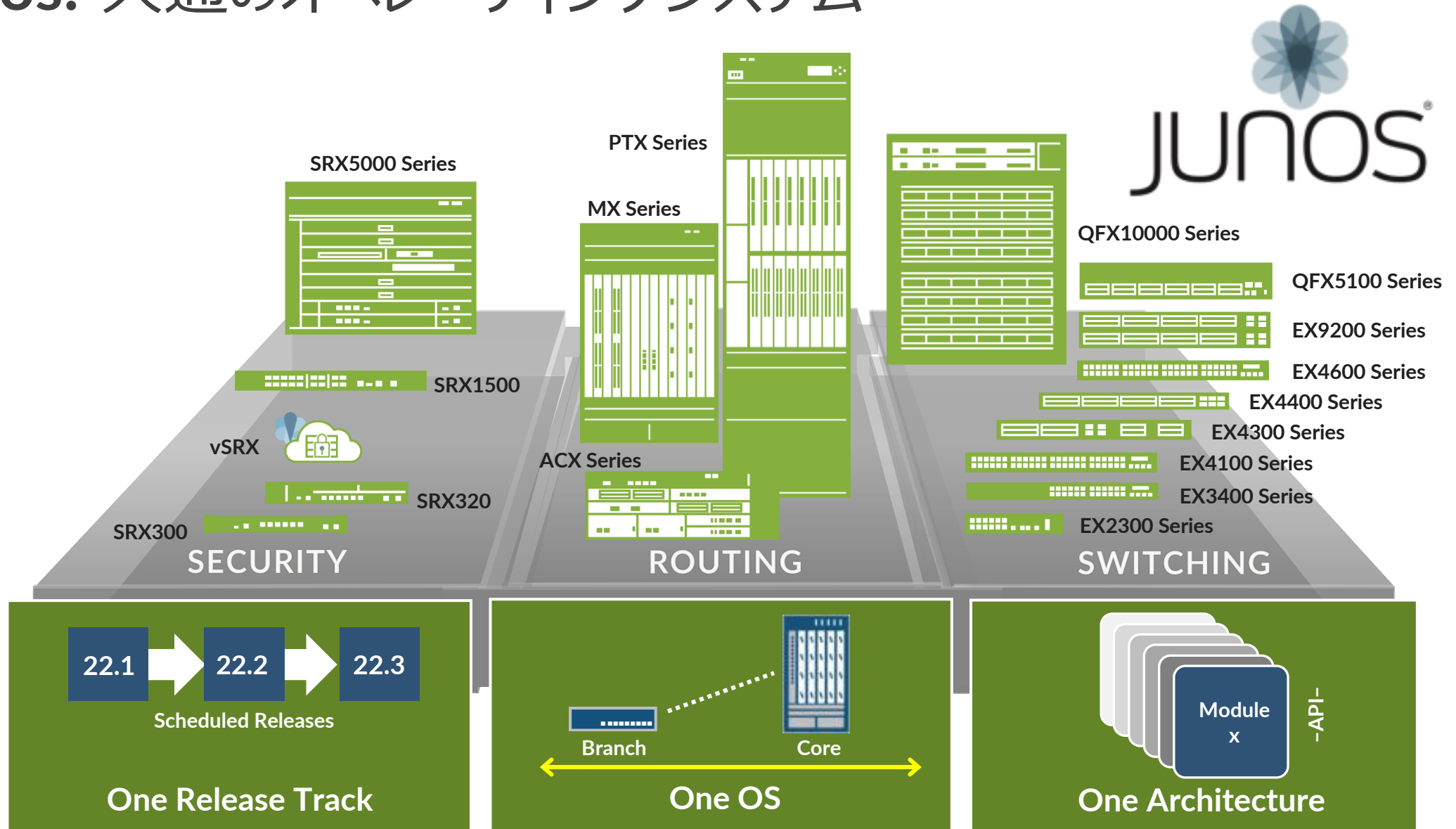
プラットフォーム毎に異なる OS と機能セット



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

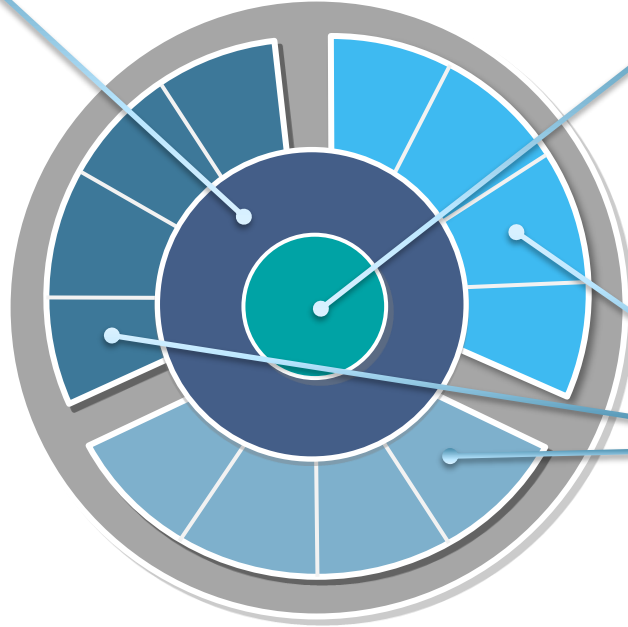
End-to-end Security

In-network Automation

SDK and Licensing of Junos

etc...


JUNOS®



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...

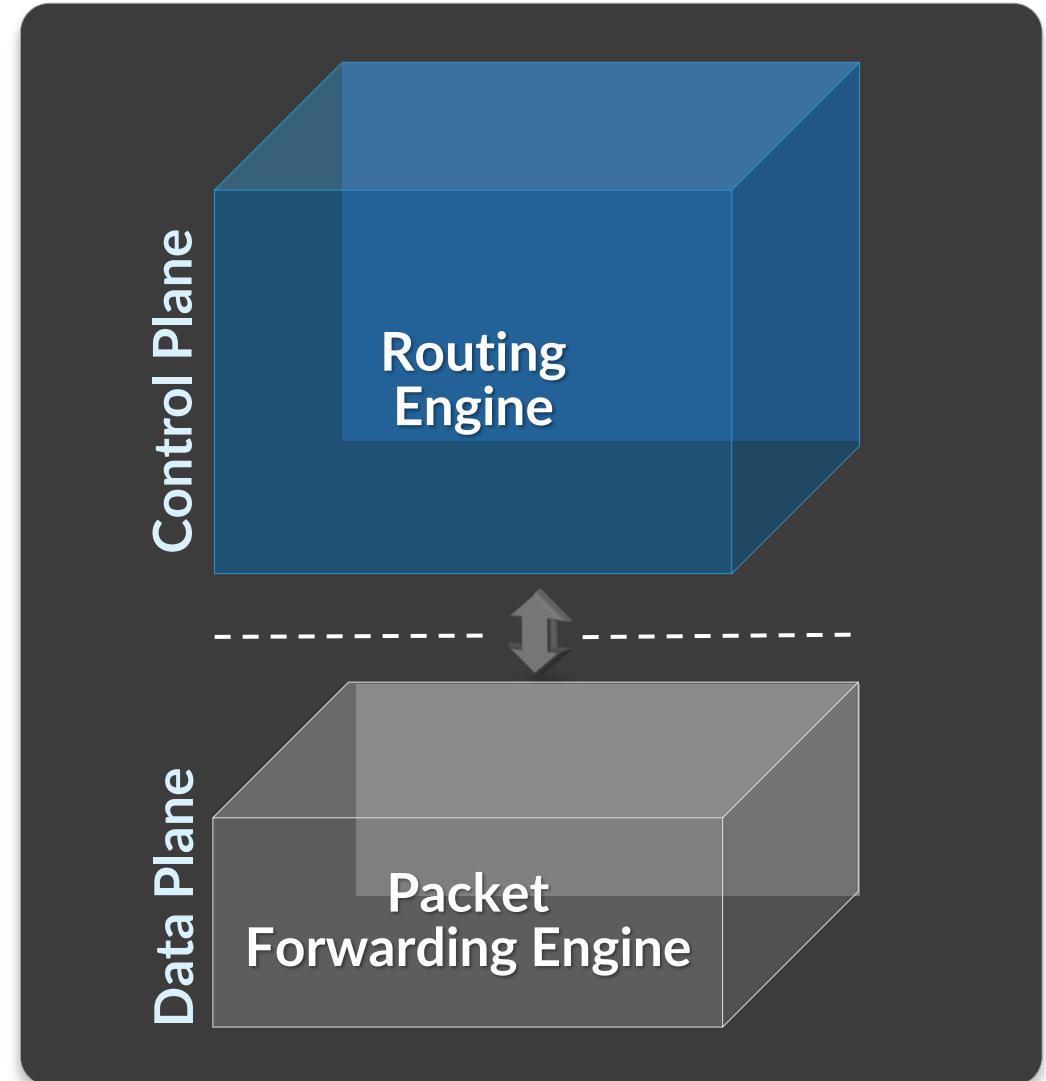
コントロールプレーンとフォワーディングプレーンの分離

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

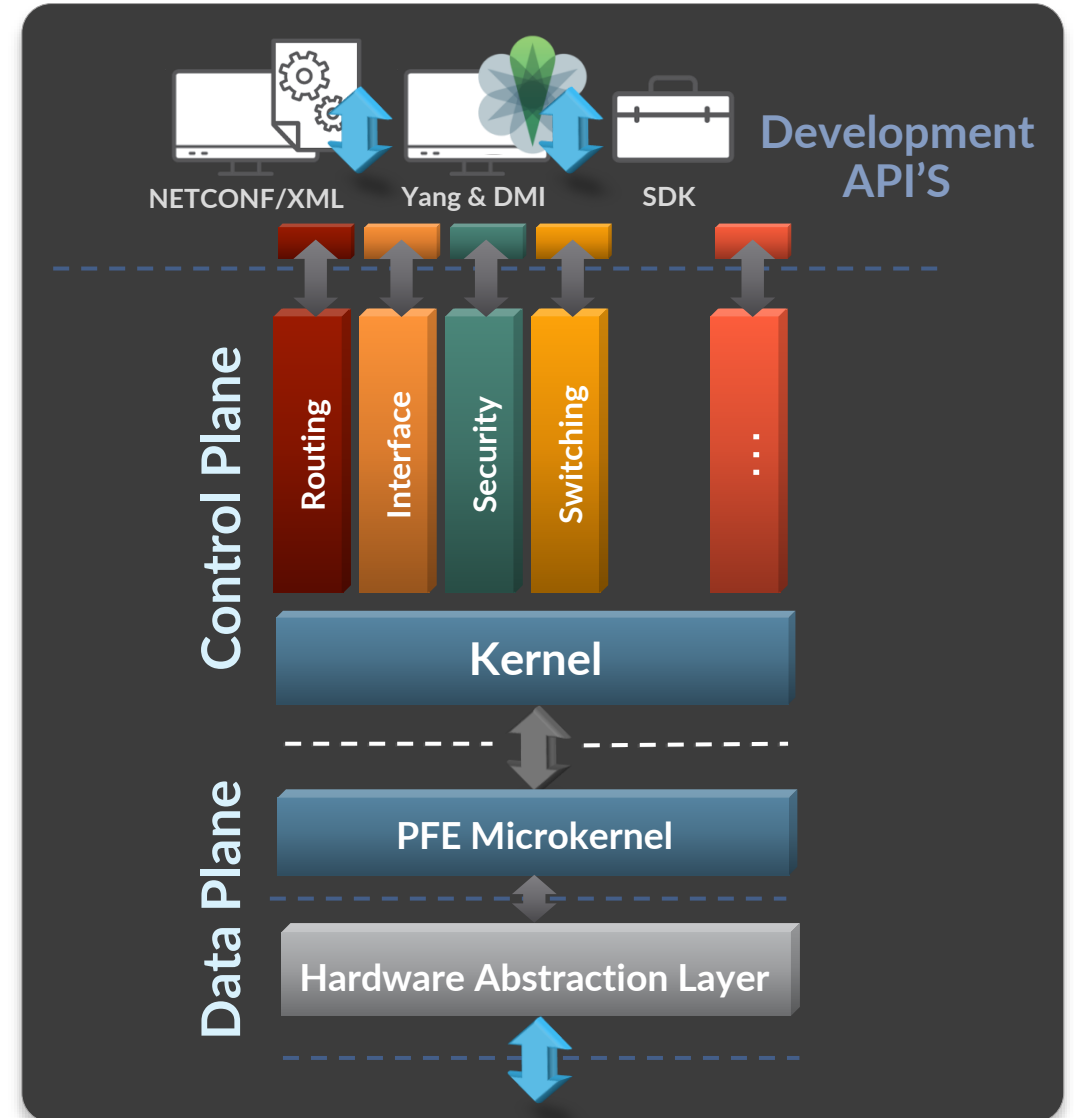
- 拡張性とパフォーマンスを担保するコンポーネント
- 冗長性、安定性、サービス拡張を効率的に提供するための独立したオペレーション

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

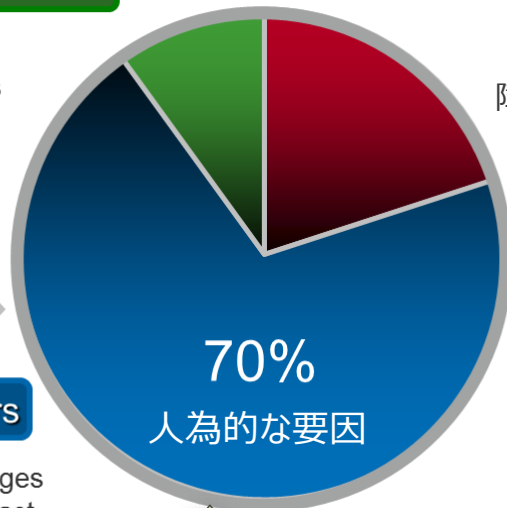
Planned Maintenance

Hardware and software upgrades

Junos Automation

Human Factors

Configuration changes that negatively impact network performance



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...

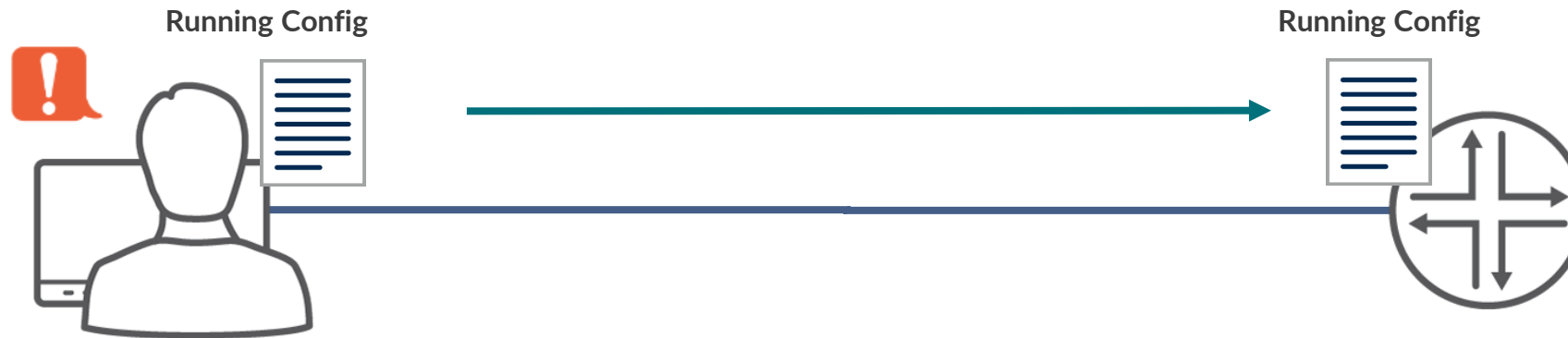
Junos CLI

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

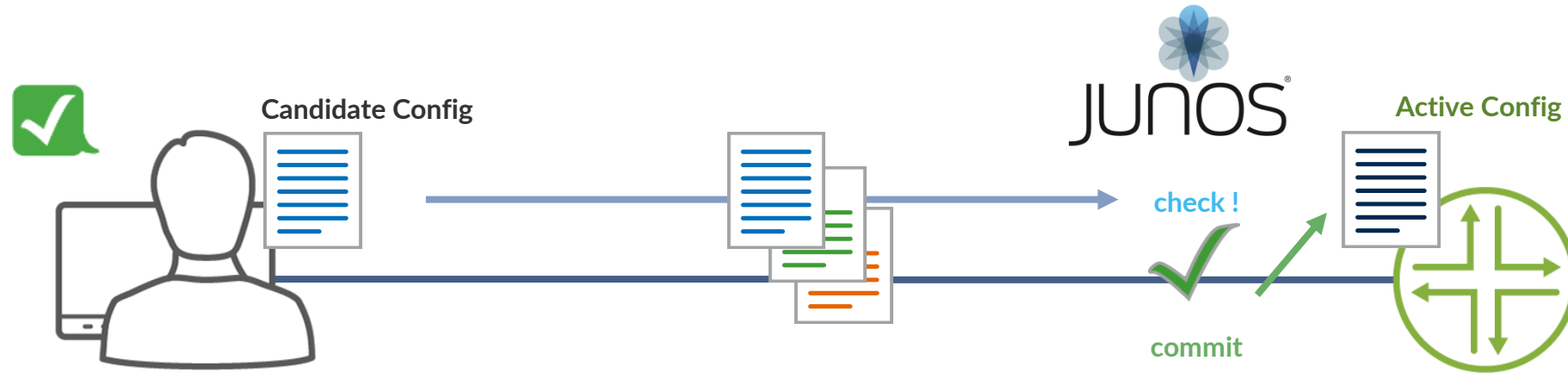
作業者の努力にたよるのではなく
ソフトウェアの仕組みでミスをなくすサポート

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ることができる**
 - 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

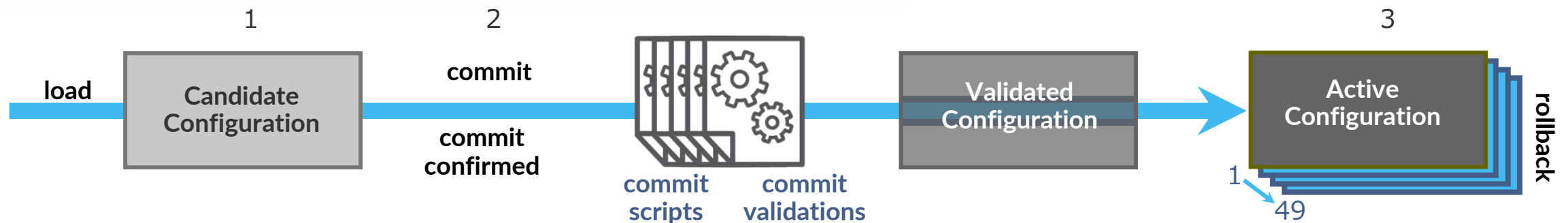
有効な Junos ツール

- “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

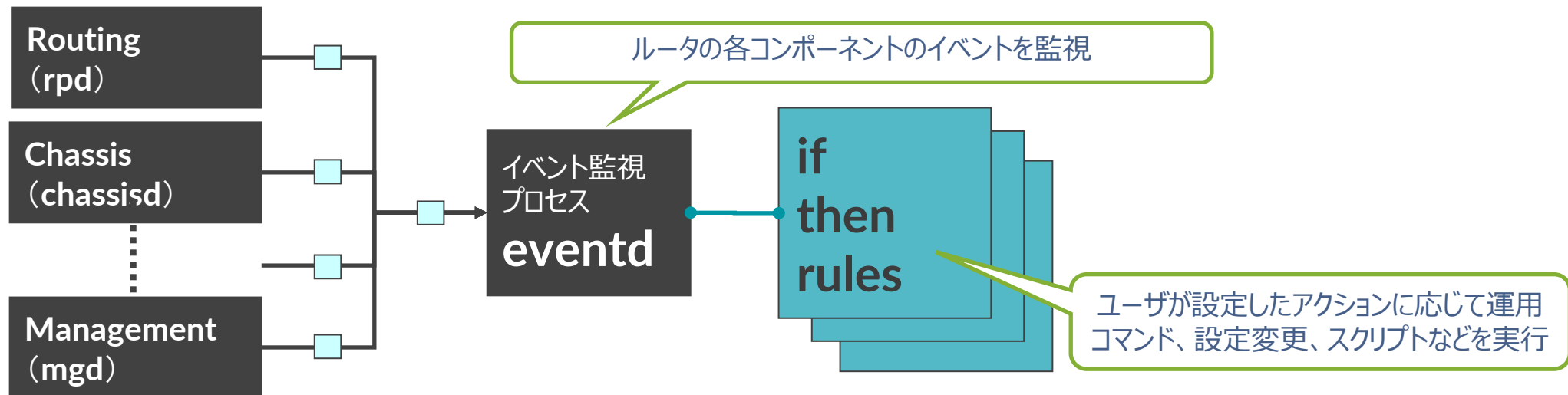
- JUNOScript とは Juniper のネットワーク装置上で動作させることができるスクリプティング機能
- Junos 自体に手を加える必要がないため、Junos の安定性を損なうことなく、ユーザ個別の自動化に対する要望に対し柔軟かつ速やかに対応することが可能
- 大別すると、運用者が起動するスクリプト “Commit Script”、“Op Script” とシステムが起動するスクリプト “Event Policy”、“Event Script” が存在

XSLT / SLAXベースのスクリプト



Junos: Event Policy/Script

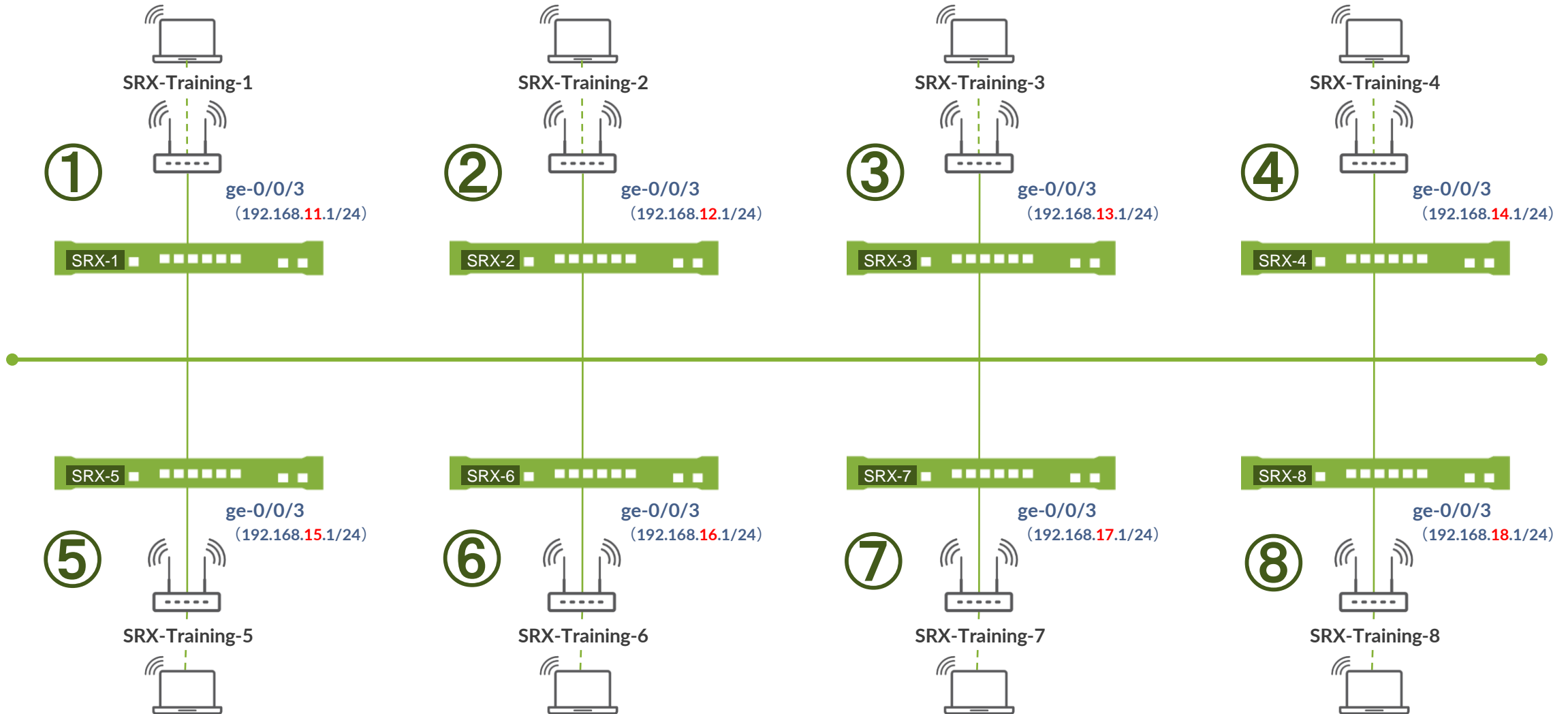
- ネットワーク機器上のイベントやタイマーをトリガーとして、コマンドやスクリプトを実行することで、運用の自動化が可能
 - イベントをトリガーとしたアクションを実行 (**Self-monitor**)
 - ルータ上の特定のイベントをトリガーとして、コマンドやスクリプトを実行
 - タイマーをトリガーとしたアクションの実行
 - インターバル設定や日時指定に応じて、コマンドやスクリプトを実行





トレーニング・デバイスへの アクセス方法

Security "SRX" Course Topology (Lab.1 : 基本操作)

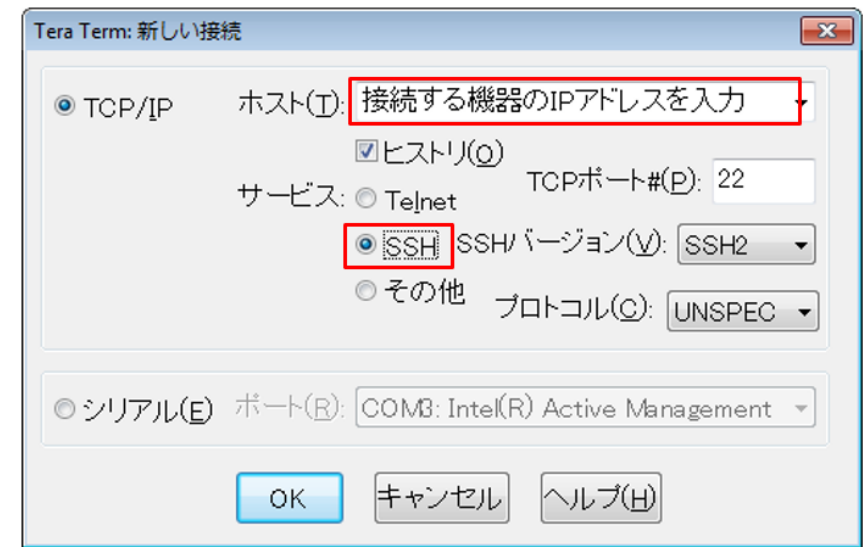


SRX ログイン

- 初期設定状態の **SRX** 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 :	192.168.1x.1
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>
```

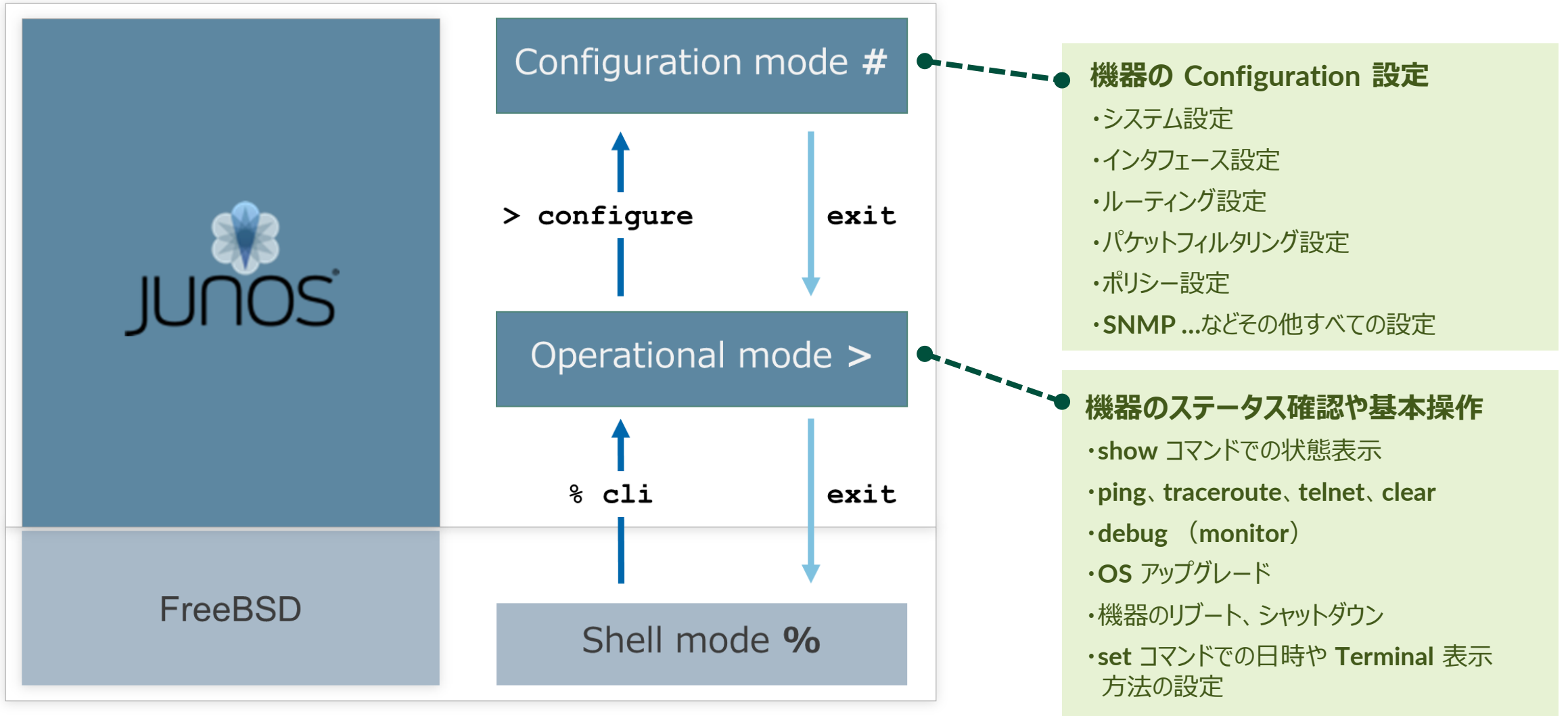




CLIモードと各モード間の移動

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
%
```

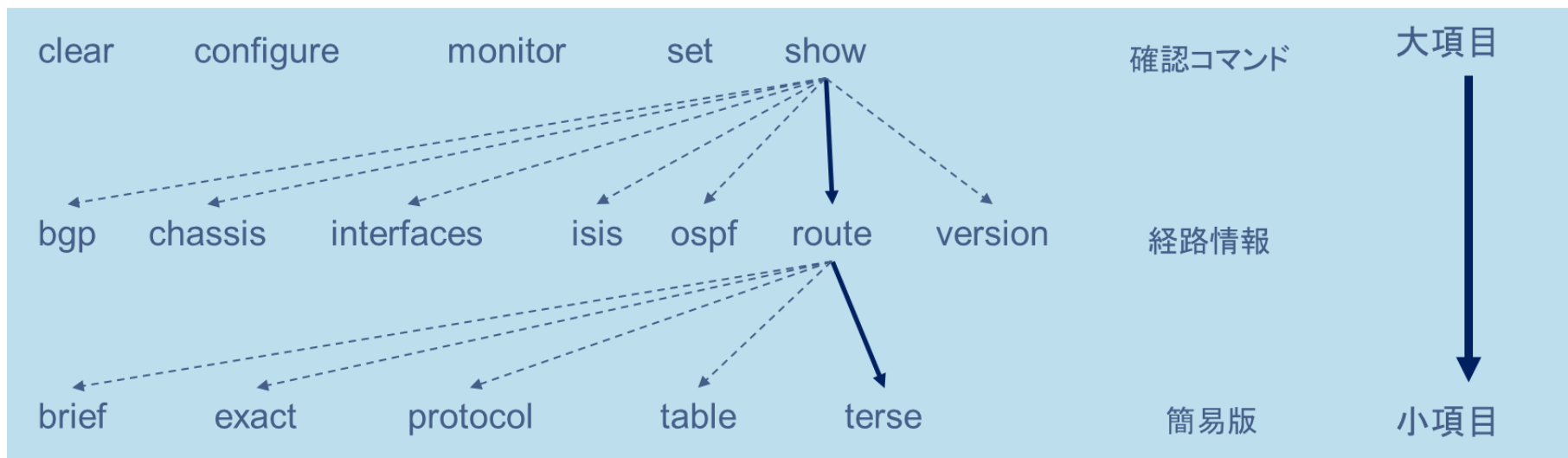
Operational モード

- 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様に表示

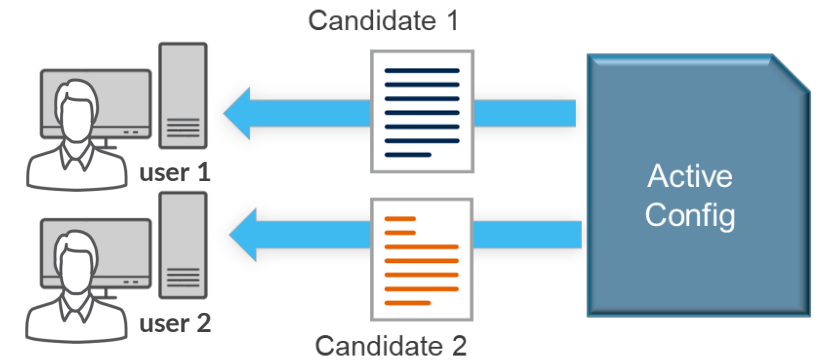
```
user> configure
Entering configuration mode
Users currently editing the configuration:
  user terminal u0 (pid 6898) on since 2022-07-15 09:15:04 UTC, idle 00:05:48
  commit-at
The configuration has been changed but not committed

[edit]
user#
```

Configuration モード： オプション

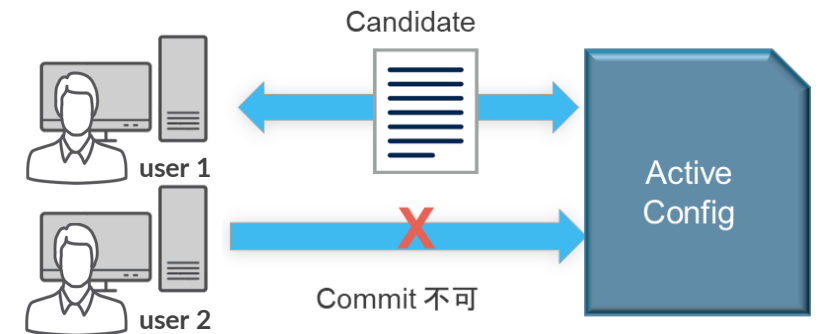
- **configure private** コマンドを使用すると、ログインユーザー専用の **Candidate Configuration** が用意される

```
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: 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
```

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:65:80, Hardware address: ec:13: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          Seconds
  Bit errors            0
  Errored blocks        0
Ethernet FEC statistics Errors
  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: Sendbcast-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, Runts: 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
  FCS 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
    FIFO errors 0
    MAC control frames 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
    % bandwidth % 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: Sendbcast-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$7IFu12bzwmnRtLm40E9546HZ6Dgkty6wfaYefYRqgd1AI
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)    0                   (0)
fti0           Up    0                   (0)    0                   (0)
gre            Up    0                   (0)    0                   (0)
ipip           Up    0                   (0)    0                   (0)
irb            Up    0                   (0)    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

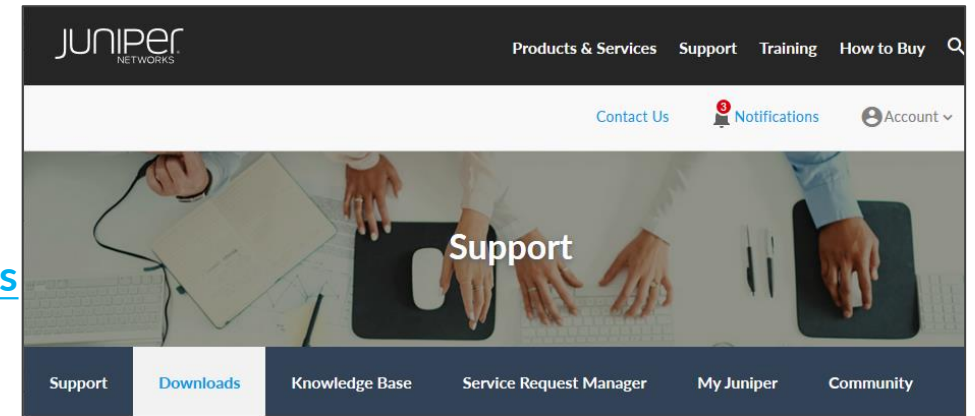
```
> file copy ftp://ログインID @アドレス/Junos パッケージ名/var/tmp
```

- 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)
```

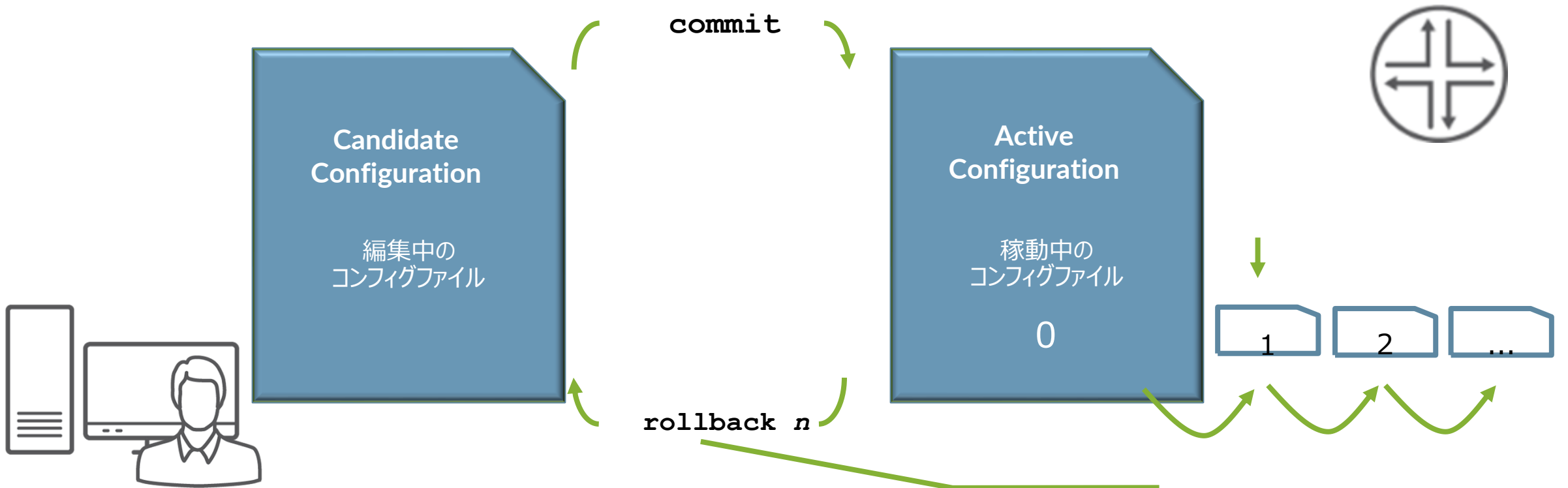
```
root> request system reboot
```



Junos CLI 操作 ～ Configuration モード ～

“Commit & Rollback”

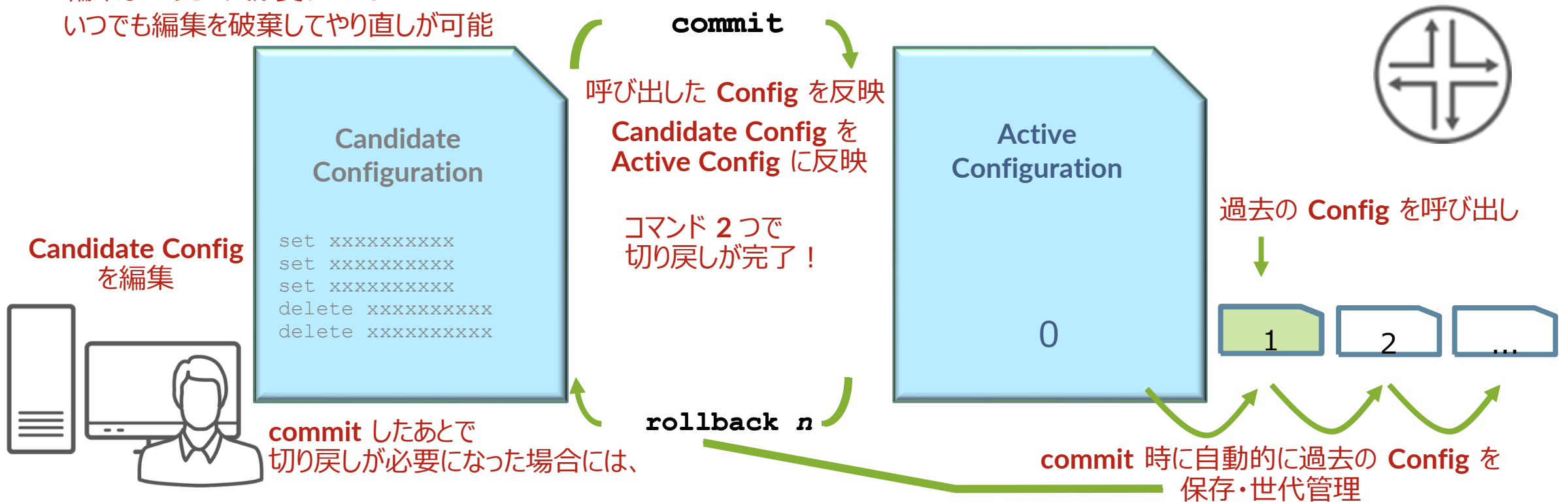
Configuration モードで行った設定変更は、Candidate Configuration として保持され、“commit” するまで設定は Active Configuration として反映されない
万一間違えた場合でも、“rollback” コマンドにてすぐに前の状態に戻ることが可能



“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

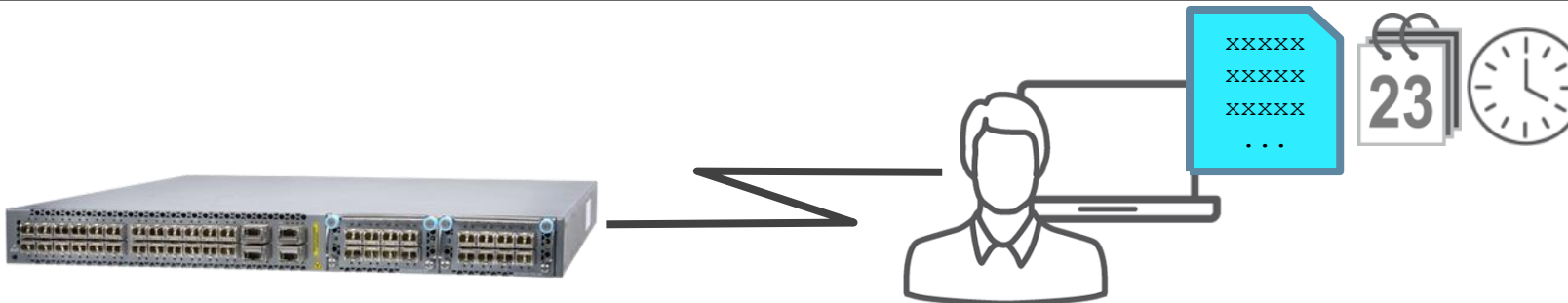


Junos : commit at time オプション

- 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]
```



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
 - 管理インターフェース設定
 - 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

- 管理インターフェース設定
 - 例 1：EX の管理インターフェース（me0）を設定

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

- 例 2：MX、SRX の管理インターフェース（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

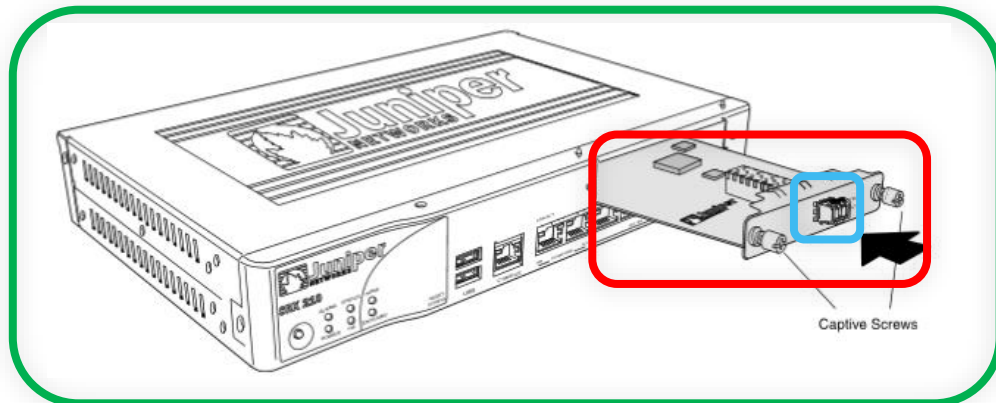
- その他のインタフェース
 - ae : LAG インタフェース
 - lo0 : Loopback インタフェース
 - me0 : EX、QFX シリーズの管理インタフェース
 - fxp0 : SRX、MX シリーズの管理インタフェース

PIC と FPC

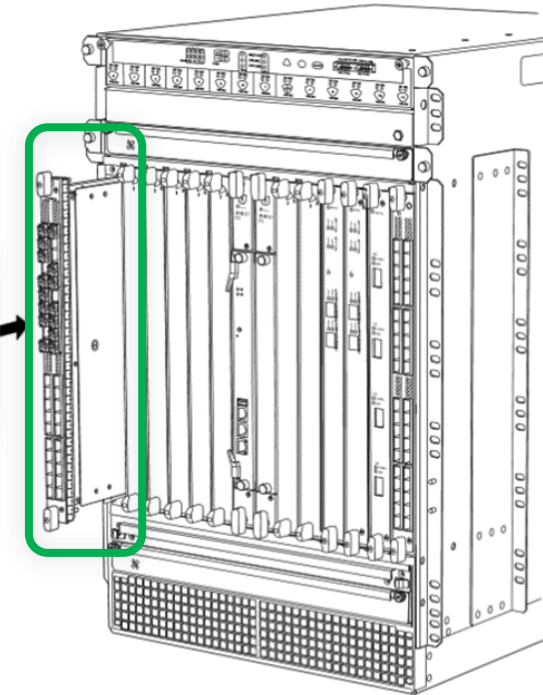
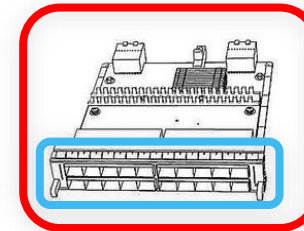
- FPC は BOX 型の筐体番号、Chassis 型のラインカード番号に相当
- PIC は FPC に接続されるアップリンクモジュールを指す

xx - X / X / X FPC PIC Port

BOX 型



Chassis 型



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

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**

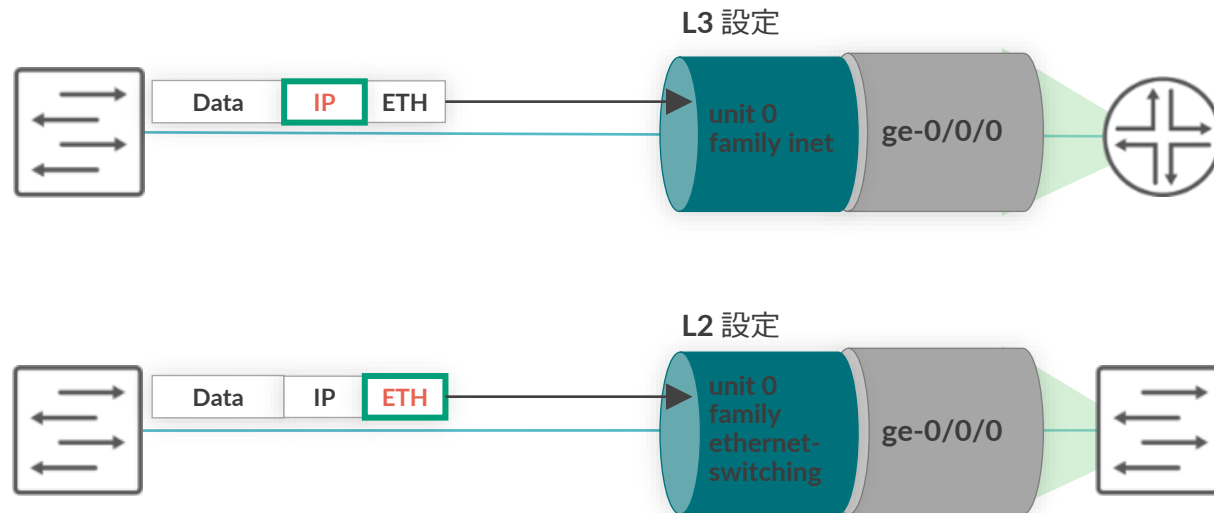
```
interfaces {  
    interface-name {  
        physical-properties;  
        [...]  
        unit unit-number {  
            logical-properties;  
            [...]  
        }  
    }  
}
```

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

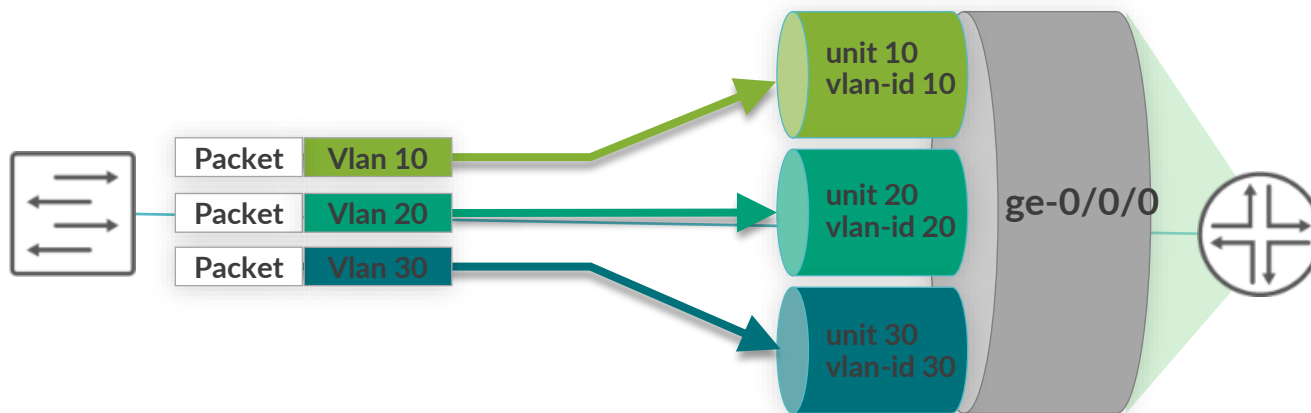


```
ge-0/0/0 {  
  unit 0 {  
    family inet {  
      address 192.168.1.1/24;  
    }  
  }  
}
```

```
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; ← admin（オペレータ）の強制的な
             インタフェースのダウン
  unit 0 {
    family inet {
      address 10.10.10.1/24;
```

```
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;  
  }  
}
```

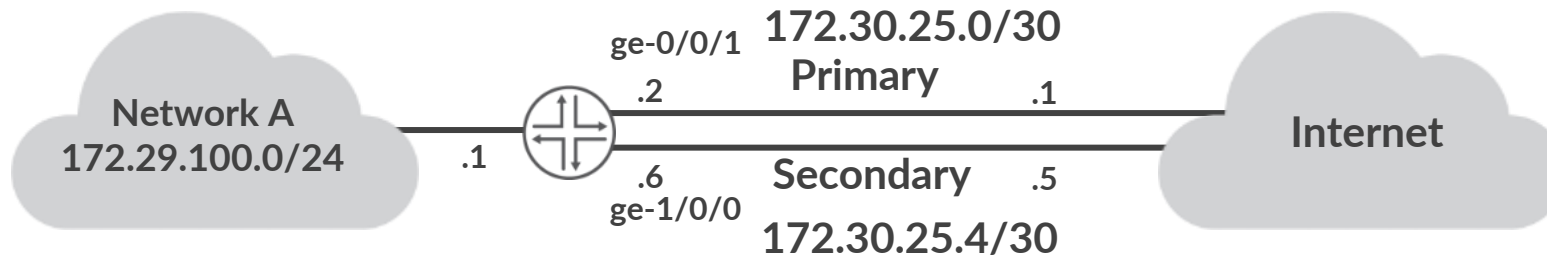
IPv4 デフォルトルートの設定

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



JUNOS Hands On Training “SRX” Course

LAB (後半)

トレーニング概要「サービスゲートウェイ “SRX” コース」

トレーニング内容（後半）	記載ページ
Juniper SRX シリーズ製品紹介	P. 95
LAB.1 Junos の基本的な操作・設定	P. 101
LAB.2 Firewall の設定	P. 114
LAB.3 NAT の設定	P. 135
LAB.4 Chassis Cluster の設定	P. 155
Appendix	P. 192



Juniper SRX シリーズ製品紹介

SRX の広範囲なセキュリティサービス

次世代
ファイアウォール

アプリケーションの
制御と可視化

不正侵入防御 (IPS)

ユーザーベース
ファイアウォール

Unified Threat Management
(既知の脅威に対する対策)

アンチウイルス

Web / コンテンツ
フィルタリング

アンチスパム

脅威インテリジェンス
プラットフォーム

ボットネット/ C&C

GEO-IP

カスタムフィード, APT

クラウドベースの
高度な脅威防御
(ゼロデイ対策)

サンドボックス

サンドボックスを
回避するマルウェア

豊富なレポートと
分析機能

SRX 基本サービス

ファイアウォール

NAT

VPN

ルーティング

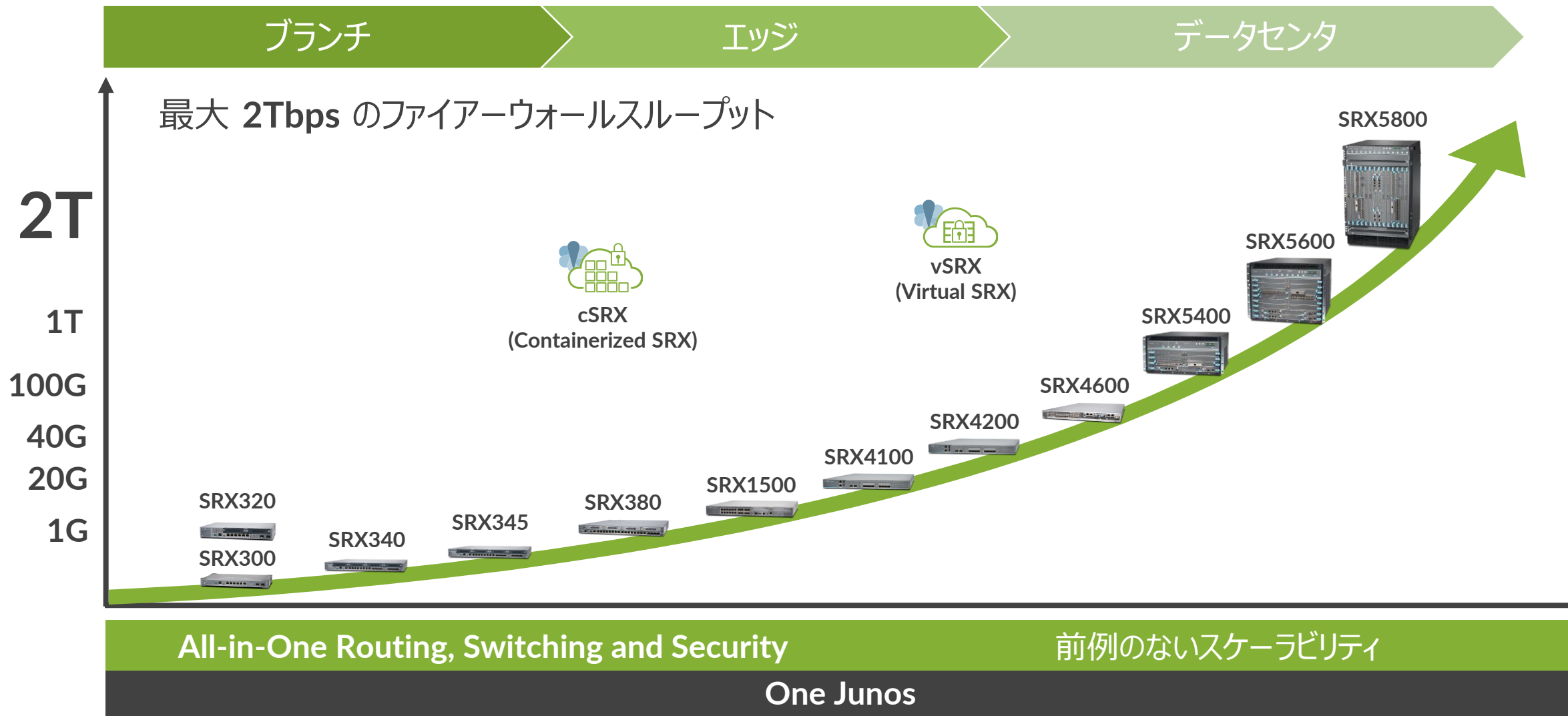
マネージメント

レポート

分析

オートメーション

製品ラインナップ



SRX の特徴



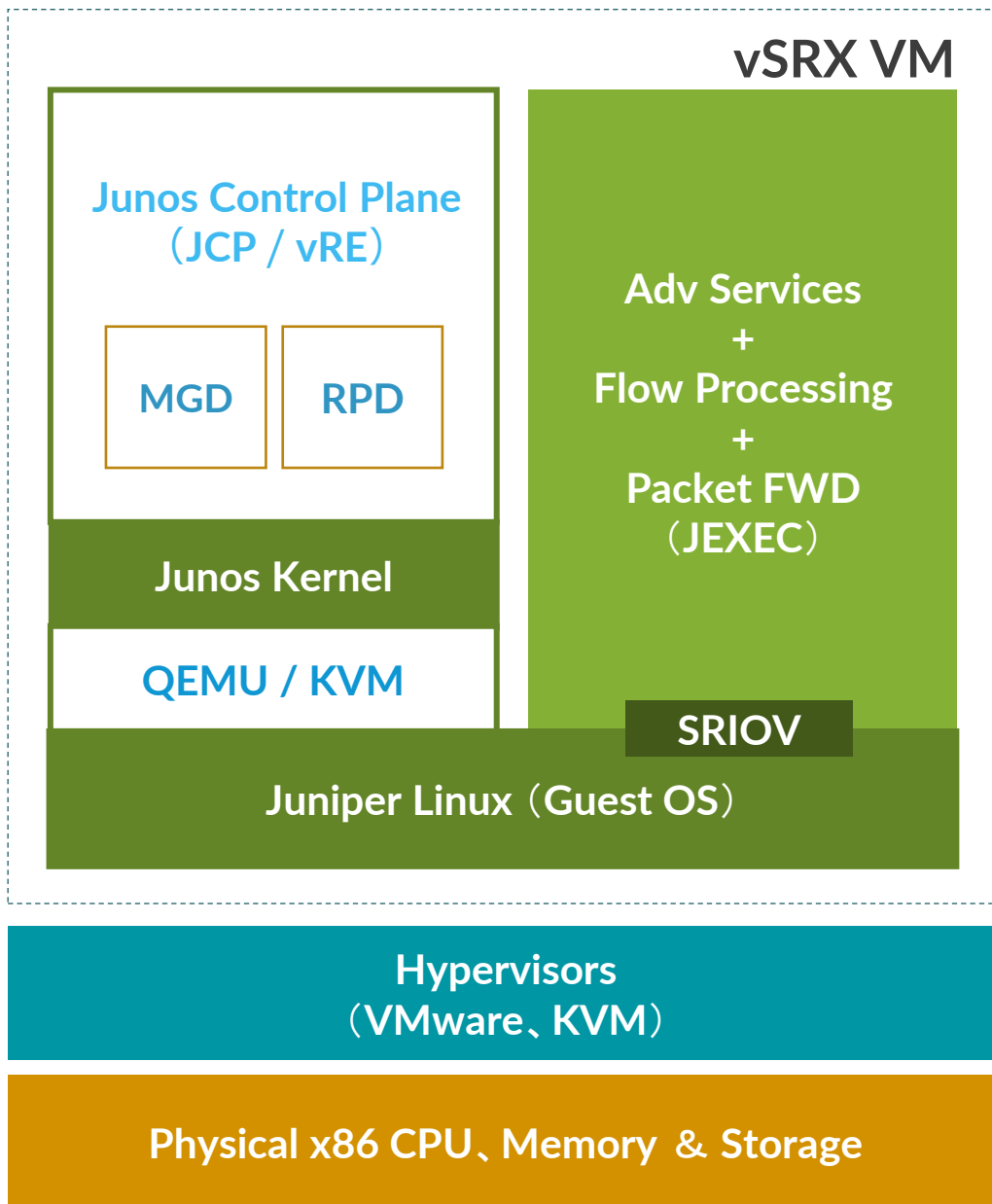
ルーティング、セキュリティ、スイッチを 1 つの筐体で実現



MAC-sec、IP Sec、およびアプリケーションセキュリティ等、全てのレイヤーへのセキュリティ



最高のエンドユーザーアプリケーションエクスペリエンスを提供



vSRX (Virtual SRX)

- ✓ **HW アプライアンス SRX と同等の機能実装**
 (Including Firewall、AppSecure、UTM / IDP、Integrated User Firewall、SSL Proxy、VPN、NAT、Routing、HA Cluster、etc.)
- ✓ **サポートプラットフォーム**
 - VMware
 - CentOS (KVM)
 - Ubuntu (KVM)
 - Contrail
- ✓ **vSRX キー・ハイライト**
 - 物理 SRX と同一の使用感で操作できる仮想ファイアウォール
 - VMware や KVM などのハイパーバイザをサポート
 - 2 vCPU で、最大約 17Gbps のファイアウォールスループットを実現
 - 業界屈指のパフォーマンス
 - vCPU を最大 12 個使用することにより、最大 100Gbps を超えるスループットを実現
 - AWS などのクラウドサービスにも対応

ハイエンド SRX シリーズサービスゲートウェイ

重要度の高い資産(リソース)に更なる厳格なセキュリティを



業界最高クラスのセキュリティ



業界最高クラスのパフォーマンス性能



キャリア・クラウド事業者にて実績多数

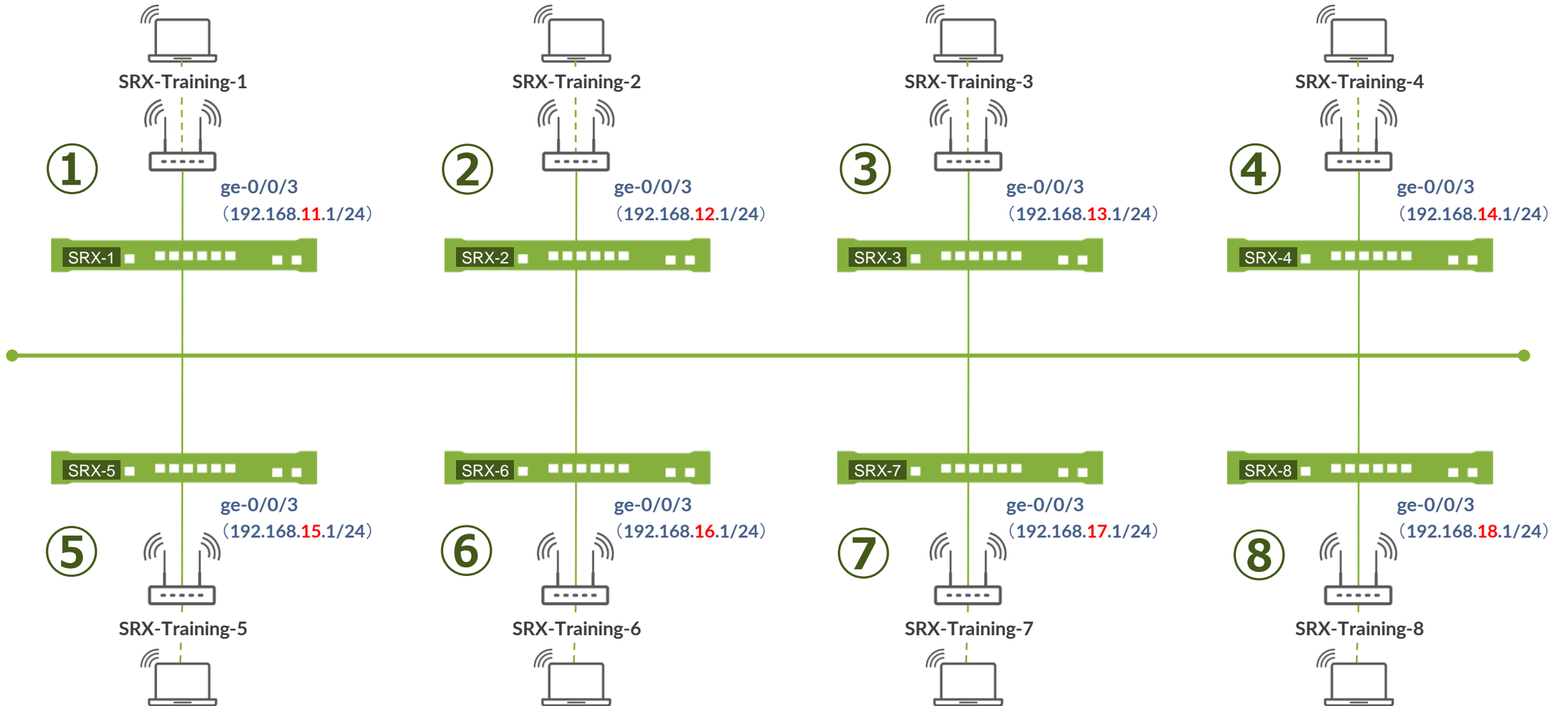




LAB.1

Junos の基本的な操作・設定

Security "SRX" Course Topology (Lab.1: 基本操作)



SRX ログイン

- 初期設定状態の **SRX** 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 :	192.168.1x.1
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)

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 SRX-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

Telnet によるアクセス

- Tera Term から telnet でアクセスできることを確認
 - telnet 192.168.1X.1
 - 作成したユーザ (lab) を使用してログイン

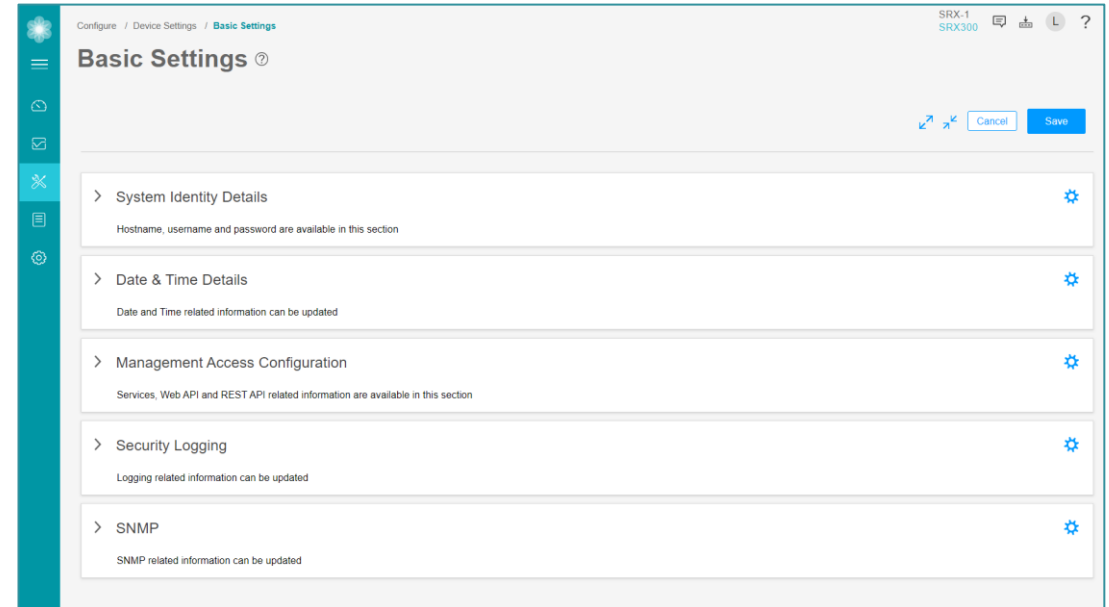
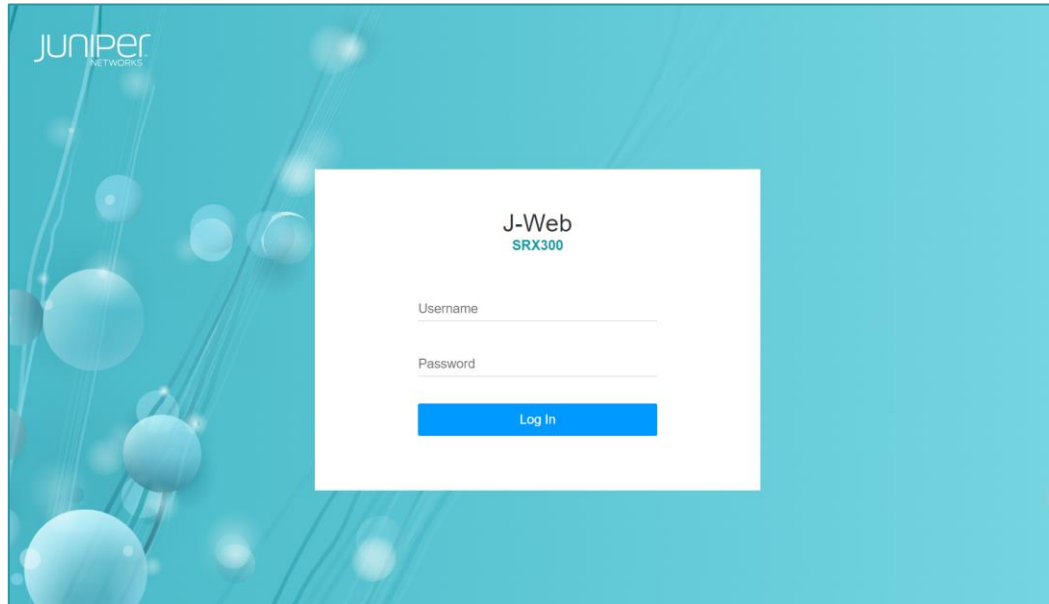
FTP によるアクセス

- 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 192.168.1X.1
 - 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 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://192.168.1X.1/
 - root、または作成したユーザ (lab) を使用してログイン

J-Web GUI



J-Web の特徴

- Junos 15.1X49-D100 から大幅に機能追加
- SRX 単体での充実したレポーティング機能
- 直感的な GUI 操作、特別なソフトウェアやライセンスは不要
- 工場出荷状態で PC と接続し、セットアップウィザードから初期設定が可能

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@SRX-1> show configuration  
lab@SRX-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@SRX-1> configure  
Entering configuration mode  
  
[edit]  
lab@SRX-1# show  
lab@SRX-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

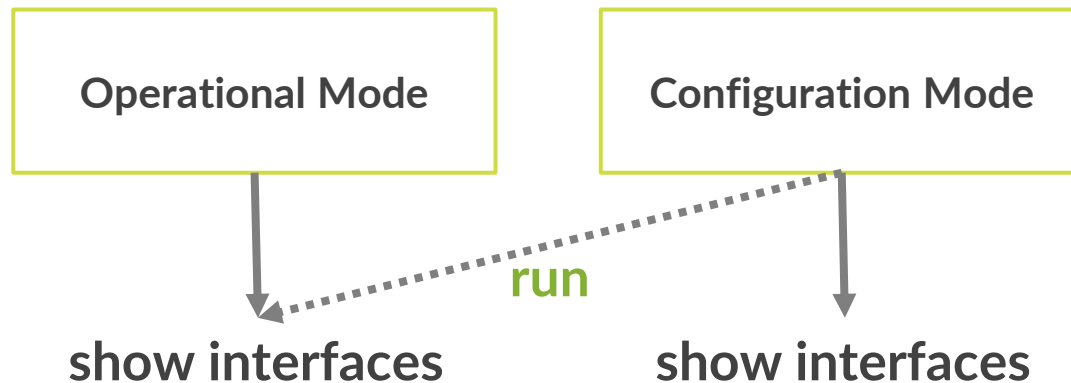
① Configuration モードにアクセス

```
lab@SRX-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@SRX-1# show interfaces  
lab@SRX-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 192.168.1X.1 -t` を継続して実行
- ② 管理インターフェースの設定を削除
インターフェースとセキュリティの設定を削除 ※`commit` はまだしないこと

```
lab@SRX-1# delete interfaces ge-0/0/3
lab@SRX-1# delete security zones security-zone trust interfaces ge-0/0/3
lab@SRX-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@SRX-1# commit confirmed 1
```

- ④ `ping` が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@SRX-1> show configuration interfaces ge-0/0/3
```

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@SRX-1# save lab1-end_YMMMDD
Wrote 76 lines of configuration to 'lab1-end_YMMMDD'

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

lab@SRX-1> file list

/cf/var/home/lab/:
lab1-end_YMMMDD
```

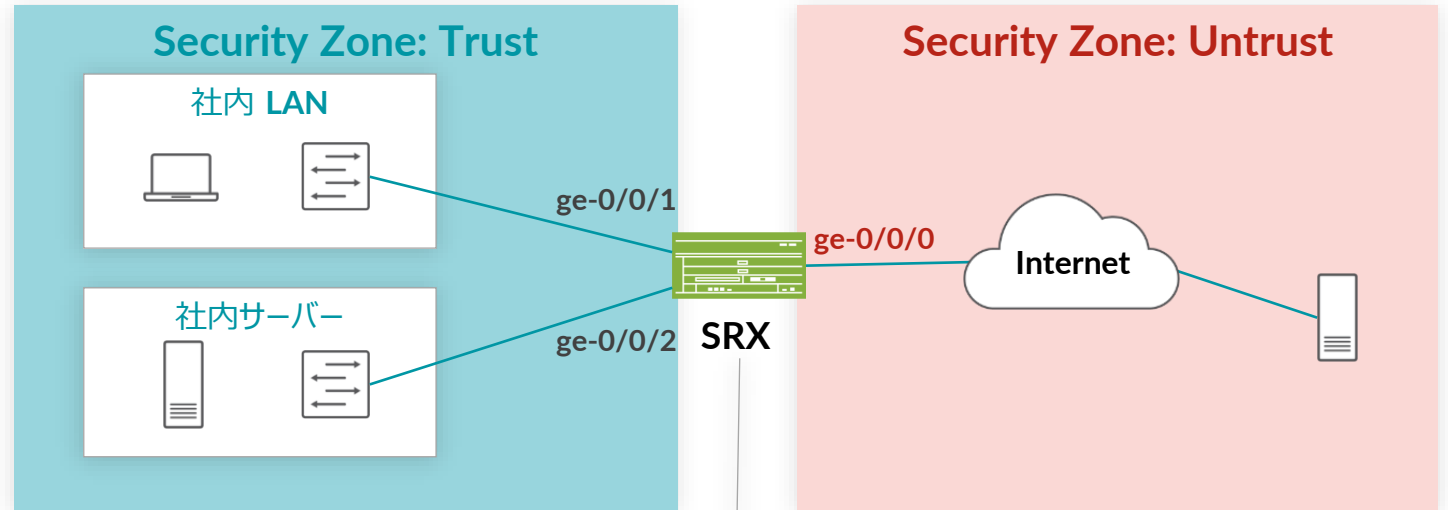



Firewall の設定

Security Zone と Policy によるトラフィック制御

- **Security Zone**

- SRX のインタフェースに割り当てる仮想的なグループ
- SRX では Security Zone を使ってトラフィックを制御



- **Security Policy**

- SRX を通過するトラフィックを制御するためのルール
- Zone 間トラフィックと、Zone 内トラフィックにそれぞれ適用される

SRX の Security Policy

Trust Zone → Untrust Zone Policy

送信元	宛先	アプリケーション	判定
社内 LAN	Any	Any	許可 (Permit)
社内サーバー	Any	Any	許可 (Permit)

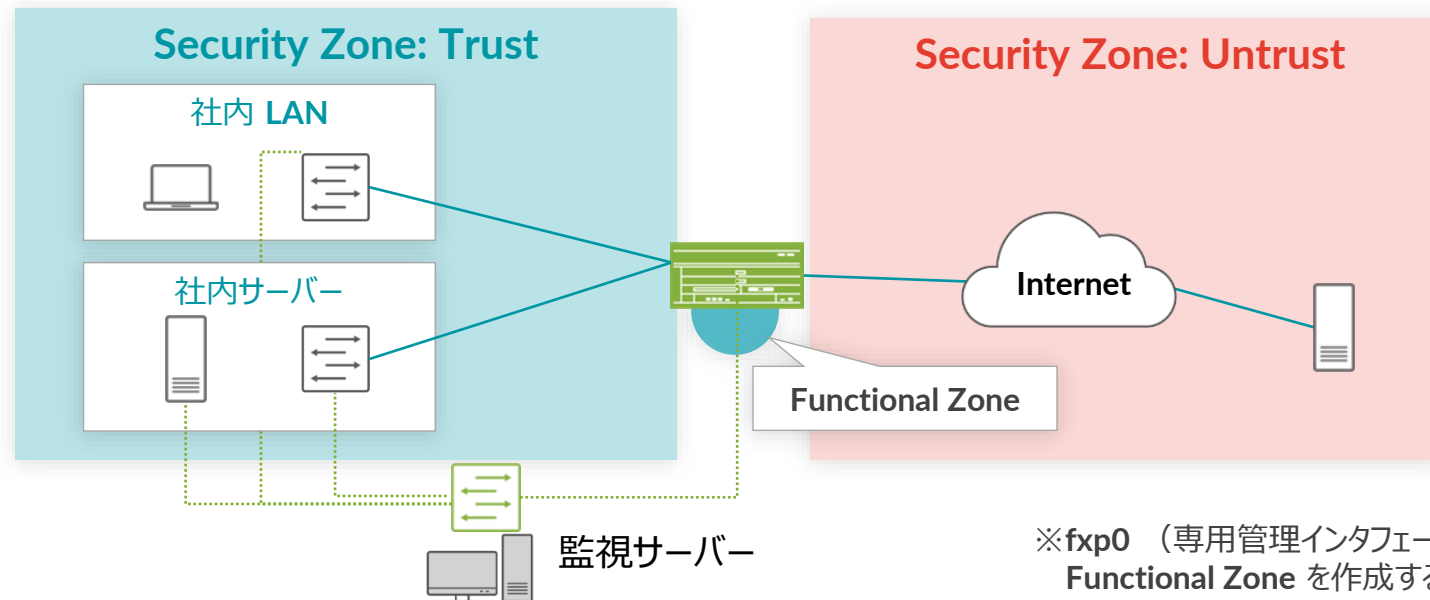
Untrust Zone → Trust Zone Policy

送信元	宛先	アプリケーション	判定
Any	社内 LAN	Any	拒否 (Deny)
Any	社内サーバー	HTTP	許可 (Permit)

Security Zone と Functional Zone

Zone には、大きく分けて下記の 2 タイプが存在

- **Security Zone**
 - SRX を通過するトラフィックを制御するための Zone
 - Security Policy は、この Security Zone 間で設定
- **Functional Zone**
 - SRX を管理するインタフェースを配置するための特別な Zone
 - この Zone で受信したトラフィックはほかの Zone に転送されない



※fxp0（専用管理インタフェース）を使用する場合は
Functional Zone を作成する必要はない

SRX で終端するトラフィックの制御

SRX で終端を許可するトラフィックを指定

- **host-inbound-traffic system-services**
 - 送受信を許可するサービスを指定
 - ftp、http、telnet、ping、etc
 - 各サービスの起動は `set system services` 配下で設定
- **host-inbound-traffic protocols**
 - 送受信を許可するプロトコルを指定
 - bgp、ospf、vrrp、etc
- **Zone 単位、または Zone 配下の Interface 単位で設定**
 - **例 1** : `trust zone` 全体で、`system-services` と `protocol` をすべて許可

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
```

- **例 2** : `untrust zone` の `interface ge-0/0/0` で、`ping` と `ospf` のみを許可

```
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
protocols ospf
```

Security Zone の設定例

- デフォルト設定

- Trust Zone

- Zone 単位で `host-inbound-traffic system-services`、`protocols` をすべて許可
 - インタフェース `irb.0` がバインド

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces irb.0
```

- Untrust Zone

- Zone 下の interface `ge-0/0/0` で `dhcp`、`tftp`、`https`、`ge-0/0/7` で `dhcp`、`tftp` を許可

```
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services dhcp
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services tftp
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services https
set security zones security-zone untrust interfaces ge-0/0/7.0 host-inbound-traffic
system-services dhcp
set security zones security-zone untrust interfaces ge-0/0/7.0 host-inbound-traffic
system-services tftp
```

Security Zone の設定確認

- Zone の設定確認コマンド

```
lab@SRX> show security zones
```

```
Security zone: trust
```

```
Send reset for non-SYN session TCP packets: Off
```

```
Policy configurable: Yes
```

```
Interfaces bound: 1 ← バインドされたインタフェース数
```

```
Interfaces:
```

```
  irb.0 ← バインドされたインタフェース名
```

```
Advanced-connection-tracking timeout: 1800
```

```
Security zone: untrust
```

```
Send reset for non-SYN session TCP packets: Off
```

```
Policy configurable: Yes
```

```
Screen: untrust-screen ← 適用されている Screen 名
```

```
Interfaces bound: 2
```

```
Interfaces:
```

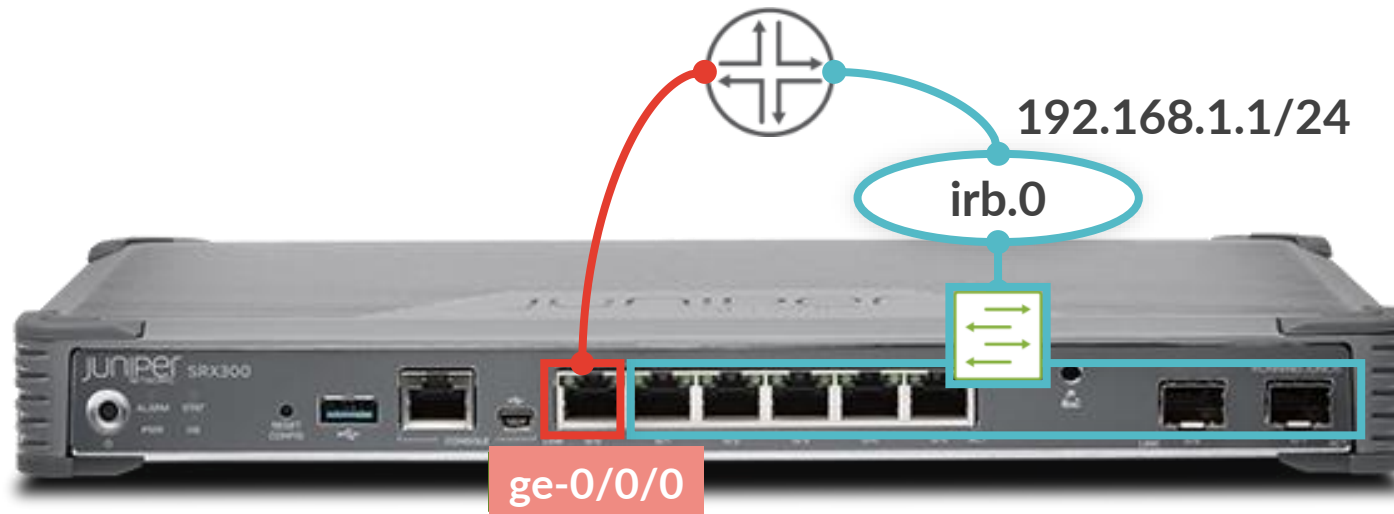
```
  ge-0/0/0.0
```

```
  ge-0/0/7.0
```

```
Advanced-connection-tracking timeout: 1800
```

補足： IRB インタフェースについて

- IRB (Integrated Routing and Bridging) とは
 - VLAN ルーティングで使用するインタフェースの名称
 - 通常のインタフェースと同様、IP アドレスをアサインして使用
- SRX300 の場合
 - ge-0/0/0 は L3 ルーティング動作
 - ge-0/0/1~7 は L2 スイッチング動作



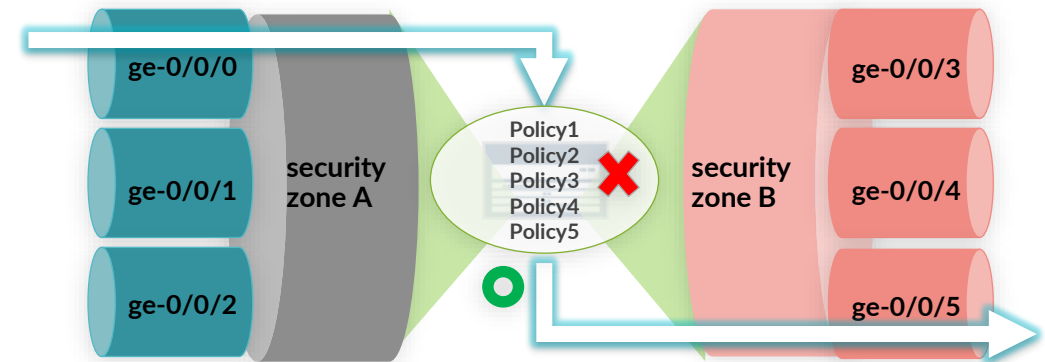
Security Policy の設定 (from-zone / to-zone)

- Policy を作成するため、送信元 Zone (from-zone) と宛先 Zone (to-zone) を定義

- Zone 間通信の Policy

例 1 : zone A から zone B への通信 Policy を作成

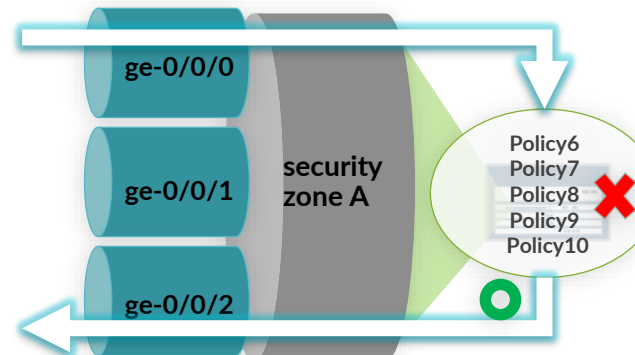
```
lab@SRX# show security policies
from-zone A to-zone B {
  policy 1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```



- Zone 内通信の Policy

例 2 : zone A から zone A への通信 Policy を作成

```
lab@SRX# show security policies
from-zone A to-zone A {
  policy 6 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```



※複数の Policy がある場合、
設定の上から順に評価される

Security Policy の設定 (match / then)

- 各 Policy では match と then でトラフィックを評価してアクションを決定
 - match - Policy に合致させる条件を設定
 - then - 条件に合致した通信に対するアクションを設定

Trust Zone → Untrust Zone Policy				
Policy 名	送信元	宛先	アプリケーション	判定
policy1	Any	Any	Any	許可 (Permit)

match

then

▼ match で指定する条件

source-address : 送信元
destination-address : 宛先
application : アプリケーション

※すべて設定必須

▼ then で指定するアクション

permit : 許可
deny : 破棄 (無応答、エラーコードを返さない)
reject : 拒否 (エラーコードを返す)
log : ログの取得
count : 該当 policy のパケット数、バイト数を取得

```
set security policies from-zone trust to-zone untrust policy policy1 match source-address any
set security policies from-zone trust to-zone untrust policy policy1 match destination-address any
set security policies from-zone trust to-zone untrust policy policy1 match application any
set security policies from-zone trust to-zone untrust policy policy1 then permit
```

Security Policy の設定 (address-book)

- address-book

- match 条件に特定のアドレスなどを指定したい場合
 - source-address / destination-address で使用したいアドレスが設定可能
 - すべてのトラフィックを指定したい場合は、予め定義されている any を使用

- address-book の設定

```
set security address-book global address AAA 1.1.1.1/32
set security address-book global address BBB 172.16.0.0/16
set security address-book global address CCC 192.168.1.0/24
```

- address-set の設定

- address-book を複数組み合わせ使用可能
- 例 : BBB と CCC を組み合わせ、BCSET を作成

```
set security address-book global address-set BCSET address BBB
set security address-book global address-set BCSET address CCC
```

- address-book の適用

```
set security policies from-zone A to-zone B policy policy1 match source-address AAA
set security policies from-zone A to-zone B policy policy1 match destination-address BCSET
```

Security Policy の設定 (default-policy)

- Policy の評価順序
 - Policy は設定の上から順番に評価される
 - match した Policy のアクションが一度だけ実行され、以後の Policy は評価されない

Trust Zone → Untrust Zone Policy				
Policy名	送信元	宛先	アプリケーション	判定
SSH	LAN	Any	ssh	許可 (Permit)
ICMP	Any	Any	icmp	許可 (Permit)
HTTPS	Any	Web	https	許可 (Permit)
default-policy	Any	Any	Any	破棄 (Deny)

上のどの Policy にも match しなければ default-policy により破棄される (変更可能)

- default-policy
 - どの Policy にも match しなかった場合に、最後に評価される Policy
 - 明示的に Policy が設定されていない場合にも使われる
 - デフォルトアクションはすべてのパケットを Drop する deny-all (暗黙の deny)
 - 以下の設定により permit-all に変更することも可能

```
set security policies default-policy permit-all
```

Security Policy の編集 (insert)

• Policy の追加

- 設定済みの Policy に新しい Policy を追加すると、最後列に挿入される
- 最後列だと評価されないケースでは順序を入れ替える必要がある

Trust Zone → Untrust Zone Policy

Policy名	送信元	宛先	アプリケーション	判定
SSH	LAN	Any	ssh	許可 (Permit)
ICMP	Any	Any	icmp	許可 (Permit)
HTTPS	Any	Web	https	許可 (Permit)
SSH_DENY	LAN	SVR_A	ssh	破棄 (Deny)

NEW!

“SSH_DENY”は
“SSH”よりも上に挿
入しないと評価され
ない

• Policy の順序入れ替え

- insert コマンドで必要な位置に Policy を並び替えることが可能
 - before policy XX policy XX の前に挿入
 - after policy XX policy XX の後に挿入

```
insert security policies from-zone trust to-zone untrust policy SSH_DENY before policy SSH
```

Screen 機能

- **Screen 機能**

- L3/L4 の基本的な攻撃防御機能を提供
- IDP モジュールを使用しない独立した機能となり、高速な処理が可能
- 攻撃防御の組 (ids-option) を作成し、それを **Zone** に適用

1 : Screen の設定を untrust-screen として作成

```
set security screen ids-option untrust-screen icmp ping-death
set security screen ids-option untrust-screen ip source-route-option
set security screen ids-option untrust-screen ip tear-drop
set security screen ids-option untrust-screen tcp syn-flood alarm-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood attack-threshold 200
set security screen ids-option untrust-screen tcp syn-flood source-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood destination-threshold 2048
set security screen ids-option untrust-screen tcp syn-flood timeout 20
set security screen ids-option untrust-screen tcp land
```

2 : 作成した untrust-screen を untrust zone に適用

```
set security zones security-zone untrust screen untrust-screen
```

Screen 機能の確認

- show security screen statistics

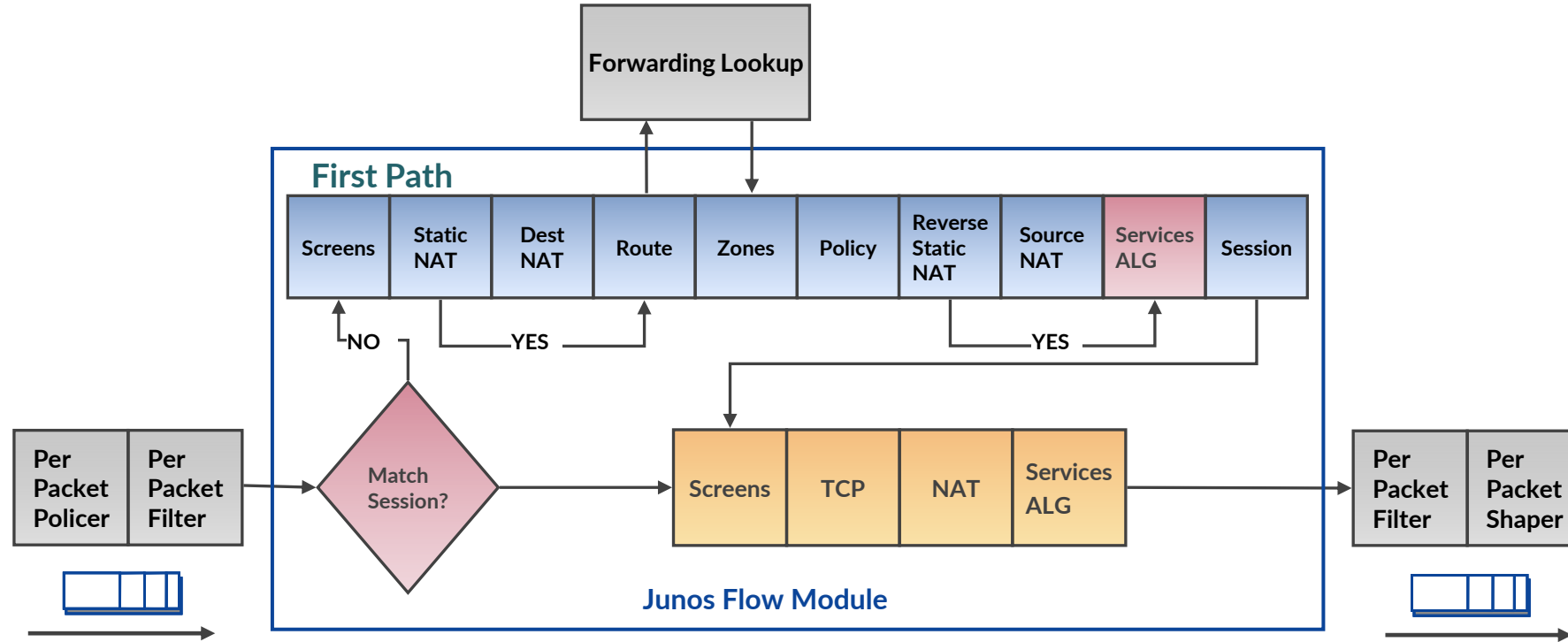
```
lab@SRX> show security screen statistics zone untrust
Screen statistics:
```

IDS attack type	Statistics
ICMP flood	5
UDP flood	205
TCP winnuke	0
TCP port scan	31
UDP port scan	0
ICMP address sweep	0
TCP sweep	22
UDP sweep	0
IP tear drop	0
TCP SYN flood	0
SYN flood source	0
SYN flood destination	0
IP spoofing	0
ICMP ping of death	0
IP source route option	0
TCP land attack	0

```
~~~~~
```

← 該当 Screen にヒットしたカウント値

SRX パケット処理の流れ (参考)



- 1) キューからパケットをピックアップ
- 2) パケット **Policy**
- 3) パケットをフィルター
- 4) セッションのルックアップ

- 5a) 新規セッションの場合
 - **FW Screen** をチェック
 - **Static、Destination NAT**
 - ルートのルックアップ
 - 宛先のゾーン **Policy** をチェック
 - **Policy** のルックアップ
 - リバース **Static、Source NAT**
 - **ALG** をセットアップ
 - セッションのインストール

- 5b) セッションが確立している場合
 - **FW Screen** をチェック
 - **TCP** をチェック
 - **NAT** トランスレーション
 - **ALG** プロセッシング

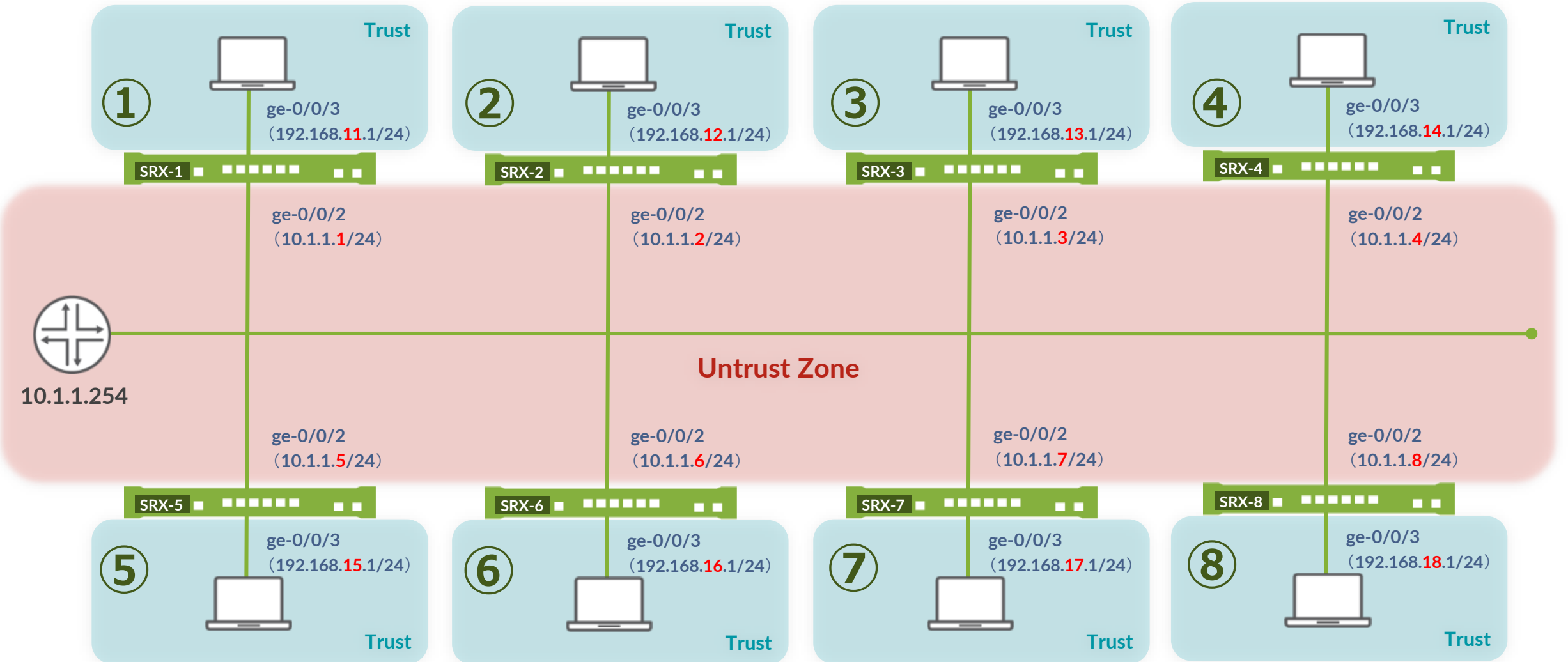
- 6) パケットをフィルター
- 7) パケットをシェーピング
- 8) パケットを転送



LAB.2

Firewall の設定

Security "SRX" Course Topology (Lab.2: Firewall の設定)



Interface、Zone の設定

- **ge-0/0/2 (Untrust 側) に IP アドレスを設定し、デフォルトルートを追加**

```
set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.X/24
set routing-options static route 0/0 next-hop 10.1.1.254
```

- **Untrust Zone を作成し、ge-0/0/2 をバインド**
 - **host-inbound-traffic で ping、telnet、ssh、http、https のサービスを許可**

```
set security zones security-zone untrust interfaces ge-0/0/2
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic system-services ssh
set security zones security-zone untrust host-inbound-traffic system-services http
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services telnet
```

Address Book の設定

- Policy で source / destination-address に使用する address-book を作成
 - Trust Zone 用

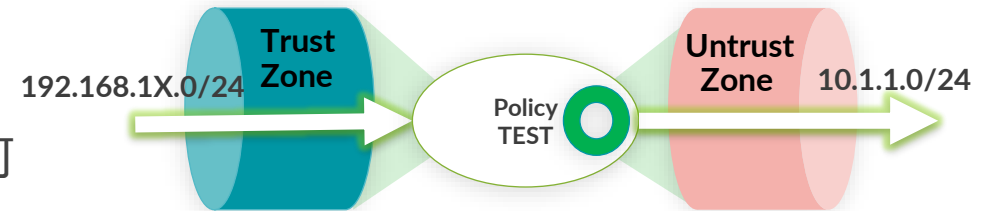
```
set security address-book global address trust-segment 192.168.1X.0/24
```

- Untrust Zone 用
 - address-set を使用

```
set security address-book global address untrust-srx 10.1.1.0/24
set security address-book global address untrust-web 192.168.1.0/24
set security address-book global address-set untrust-segment address untrust-srx
set security address-book global address-set untrust-segment address untrust-web
```

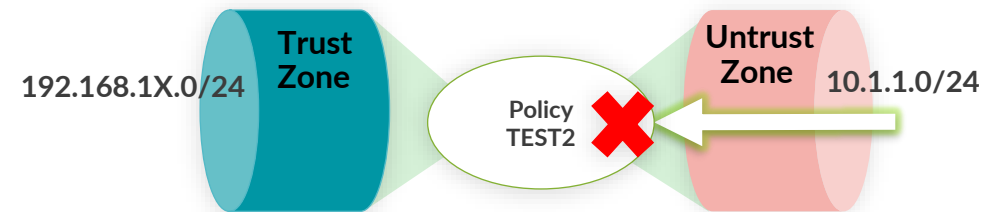
Security Policy の設定

- ① Trust Zone から Untrust Zone へのポリシーを設定
 - trust-segment から untrust-segment の通信はすべて許可



```
set security policies from-zone trust to-zone untrust policy TEST match source-address trust-segment
set security policies from-zone trust to-zone untrust policy TEST match destination-address untrust-segment
set security policies from-zone trust to-zone untrust policy TEST match application any
set security policies from-zone trust to-zone untrust policy TEST then permit
```

- ② Untrust Zone から Trust Zone へのポリシーを設定
 - すべて不許可



```
set security policies from-zone untrust to-zone trust policy TEST2 match source-address any
set security policies from-zone untrust to-zone trust policy TEST2 match destination-address any
set security policies from-zone untrust to-zone trust policy TEST2 match application any
set security policies from-zone untrust to-zone trust policy TEST2 then deny
```

```
commit
```

Security Policy の確認

• Trust から Untrust へのポリシー確認

- ① コマンドプロンプトを立ち上げ、PC から 10.1.1.254 に対して Ping を実行
 - 応答があれば正しくポリシーが動作していることが確認可能

- ② Tera Term の新規セッションで、隣の SRX の IP アドレス (Untrust) に telnet を実行

- 宛先は右側の表を参照
- ログインプロンプトが開いたら、lab / lab123 でログイン
- 自分の SRX 上で、以下のコマンドを確認
 - show security flow session
 - show security policies detail

※結果例 ⇒ Backup Slides 資料参照

• Untrust から Trust へのポリシー確認

- ③ コマンドプロンプトから隣の SRX (Trust) に Ping を実行
 - 宛先は右側の表を参照
 - Ping に応答がないことを確認

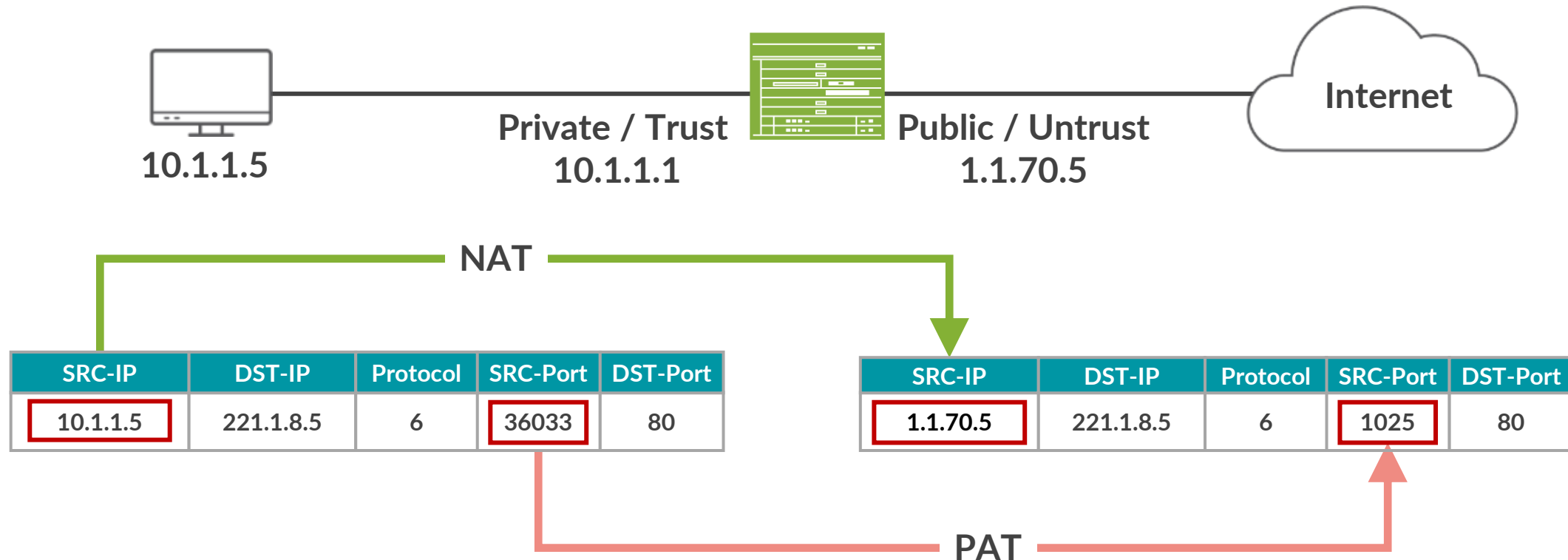
座席番号	② の宛先 (Untrust)	③ の宛先 (Trust)
1	10.1.1.2	192.168.12.1
2	10.1.1.1	192.168.11.1
3	10.1.1.4	192.168.14.1
4	10.1.1.3	192.168.13.1
5	10.1.1.6	192.168.16.1
6	10.1.1.5	192.168.15.1
7	10.1.1.8	192.168.18.1
8	10.1.1.7	192.168.17.1



NAT の設定

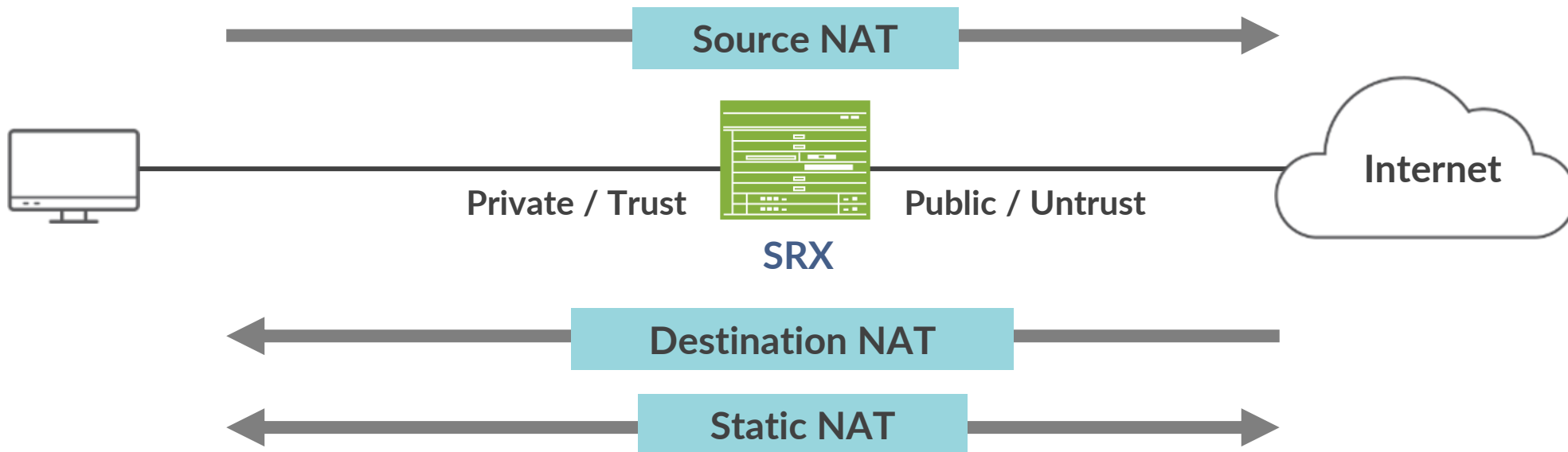
NAT 概要

- Public と Private の IP アドレスを変換
- セキュリティポリシーとは別の NAT ポリシーにて管理・設定
- ポート変換（Port Address Translation：PAT）もサポート



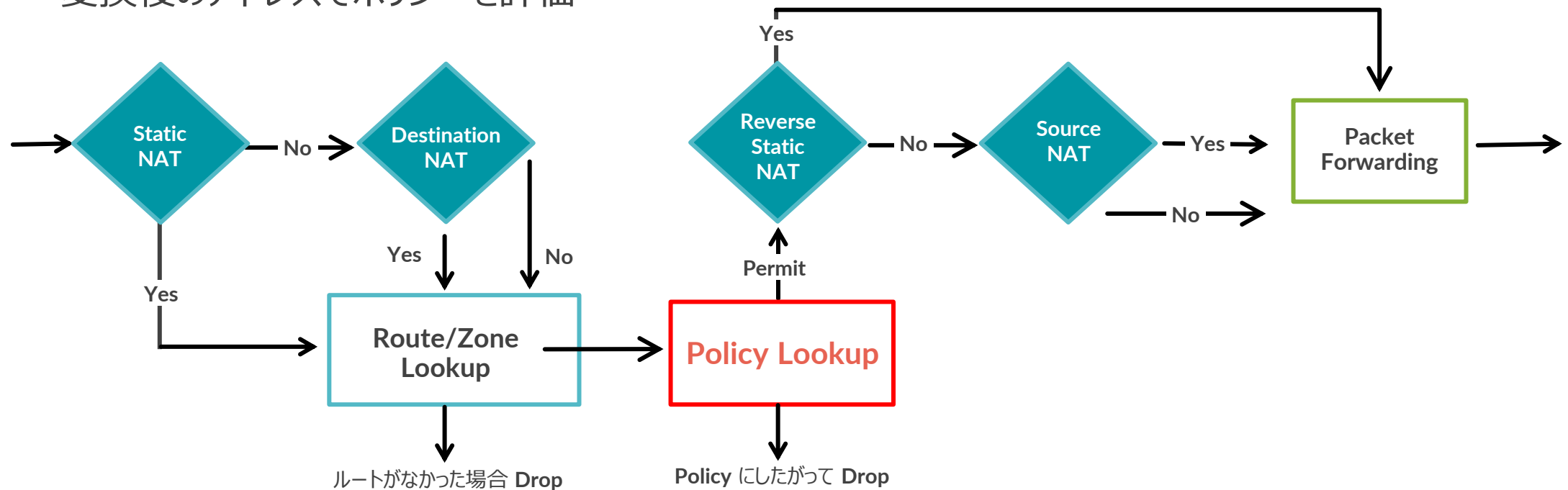
SRX の NAT タイプ

- 大きく分けて以下の 3 タイプの NAT / PAT 方法
 - Source NAT 送信元 IP アドレスを変換
 - Destination NAT 宛先 IP アドレスを変換
 - Static NAT 1 つの Private IP に 1 つの Public IP をマッピングして変換
- Source / Destination の組み合わせも可能



NAT 処理の順序

- **Source NAT**
 - セキュリティポリシー適用後に処理
 - 変換前のアドレスでポリシー評価
- **Static & Destination NAT**
 - セキュリティポリシー適用前に処理
 - 変換後のアドレスでポリシーを評価



NAT ルールの適用条件

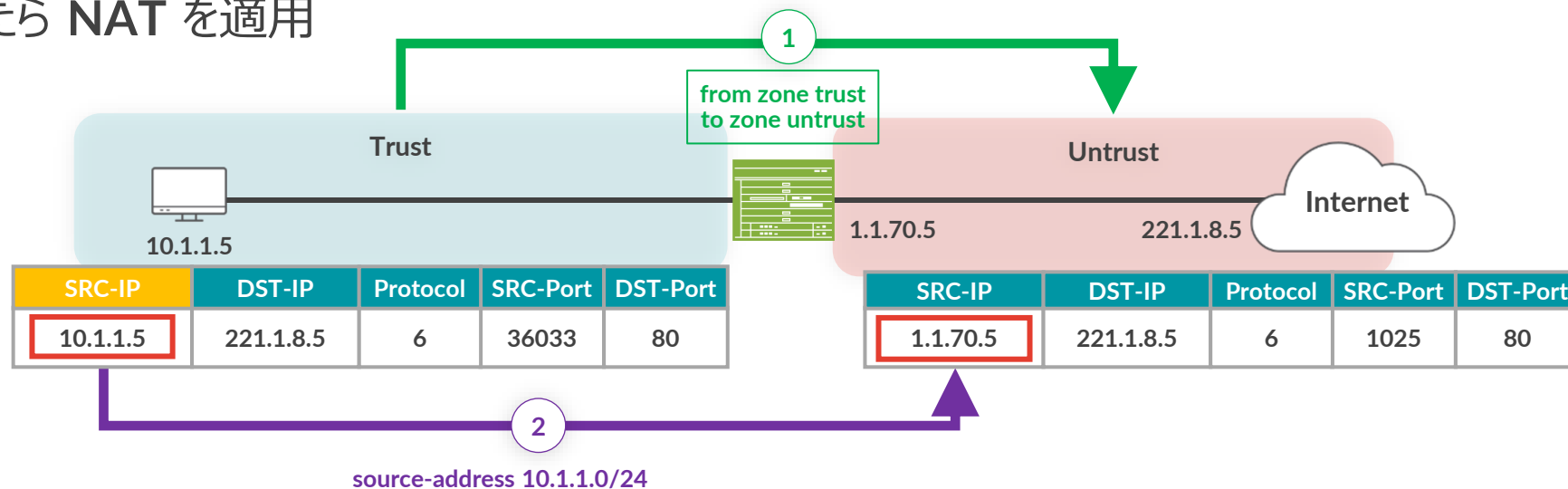
- NAT を適用するかどうか決める「2段階」

- ① 通信の方向 (rule-set)

- from – to で、「どこから」「どこへ」の通信かを指定
 - from : zone、interface、routing-instance (VR)
 - to : zone、interface、routing-instance (VR)
 - 条件にマッチしたら ② の評価に

- ② パケットの情報 (NAT Rule)

- 送信元アドレス、宛先アドレス、ポート番号を条件として「どんな」通信かを指定
 - マッチしたら NAT を適用

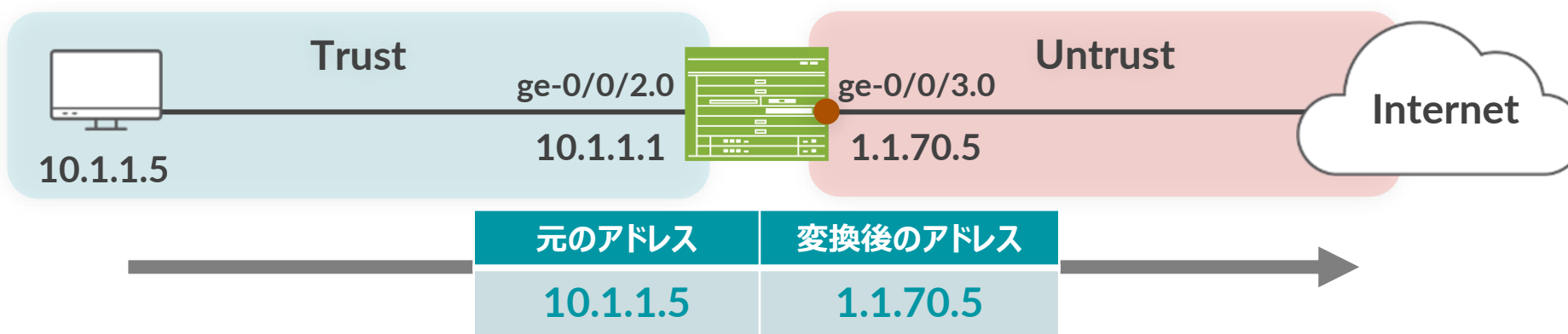


Source NAT 概要

- 送信元 IP アドレスを変換
 - オプションで送信元ポート番号の変換 (PAT)
- Source NAT の種類
 - **Interface-based source NAT**
 - SRX のインタフェースアドレスに変換
 - PAT は常時動作
 - **Pool-based source NAT**
 - Pool から IP アドレスを動的にアサイン
 - PAT はあり、なしどちらも対応

Interface-based Source NAT 設定

- Trust から入ってきたトラフィックの送信元 IP アドレスを Untrust の出口側インタフェースの IP アドレス “1.1.70.5” に変換



- NAT ルールセットで通信の方向を決定

```
set security nat source rule-set 1 from zone trust
set security nat source rule-set 1 to zone untrust
```

- NAT ルールを設定
- 送信元アドレス (0.0.0.0/0 = any) にマッチしたら、interface アドレスに変換

```
set security nat source rule-set 1 rule 1A match source-address 0.0.0.0/0
set security nat source rule-set 1 rule 1A then source-nat interface
```


Interface-based Source NAT 確認

- 変換結果の確認コマンド
 - show security flow session
 - show security nat source summary

```
user@SRX> show security flow session
Session ID: 274, Policy name: test/6, Timeout: 1794, Valid
  In: 10.1.1.5/54927 --> 221.1.8.5/23;tcp, Conn Tag: 0x0, If: ge-0/0/2.0, Pkts: 9, Bytes: 442,
  Out: 221.1.8.5/23 --> 1.1.70.5/25104;tcp, Conn Tag: 0x0, If: ge-0/0/3.0, Pkts: 10, Bytes: 483,
Total sessions: 1

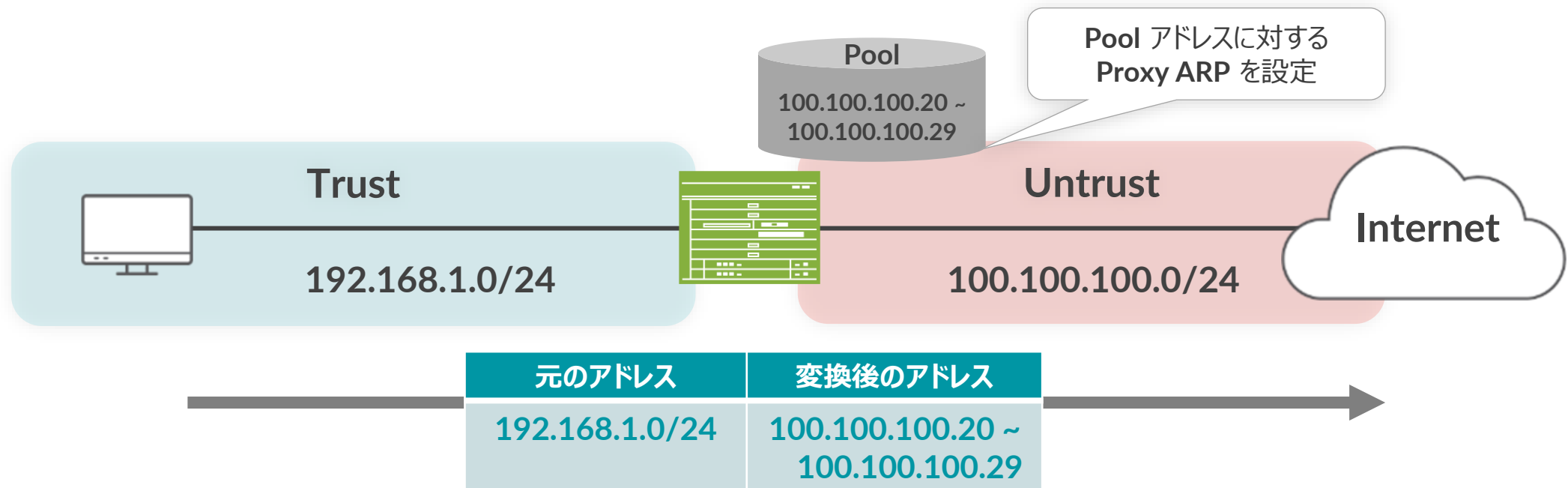
user@SRX> show security nat source summary
Total pools: 0

Total rules: 1
Rule name      Rule set      From          To            Action
1A             1             trust        untrust      interface
```

PAT も同時に動作

Pool-based Source NAT

- Trust から Untrust に抜けるトラフィックの送信元 IP アドレスを Pool アドレスに変換
- Proxy ARP を設定
 - Pool アドレスに対して SRX から ARP 応答するように設定
 - Pool アドレスとインタフェースが同じサブネット上の場合に必要



Pool-based Source NAT 設定

- アドレスプールの設定

```
set security nat source pool src_nat_pool_napt address 100.100.100.20/32 to 100.100.100.29/32
```

- NAT ルールセットの設定

- Trust ゾーンから Untrust ゾーンへの通信

```
set security nat source rule-set src_nat_napt from zone trust  
set security nat source rule-set src_nat_napt to zone untrust
```

- NAT ルールの設定

- 送信元アドレスが **192.168.1.0/24** の場合、Pool アドレスに変換

```
set security nat source rule-set src_nat_napt rule napt_1 match source-address 192.168.1.0/24  
set security nat source rule-set src_nat_napt rule napt_1 then source-nat pool src_nat_pool_napt
```

- Proxy ARP の設定

```
set security nat proxy-arp interface ge-0/0/0.0 address 100.100.100.20/32 to 100.100.100.29/32
```

Pool-based Source NAT 確認

- 変換結果の確認コマンド
 - show security flow session
 - show security nat source summary

```
user@SRX> show security flow session
Session ID: 394, Policy name: test/6, Timeout: 1794, Valid
  In: 192.168.1.26/51414 --> 100.100.100.200/23;tcp, Conn Tag: 0x0, If: ge-0/0/2.0, Pkts: 9, Bytes: 442,
  Out: 100.100.100.200/23 --> 100.100.100.20/3962;tcp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 10, Bytes: 483,
Total sessions: 1

user@SRX> show security nat source summary
Total port number usage for port translation pool: 645120
Maximum port number for port translation pool: 67108864
Total pools: 1

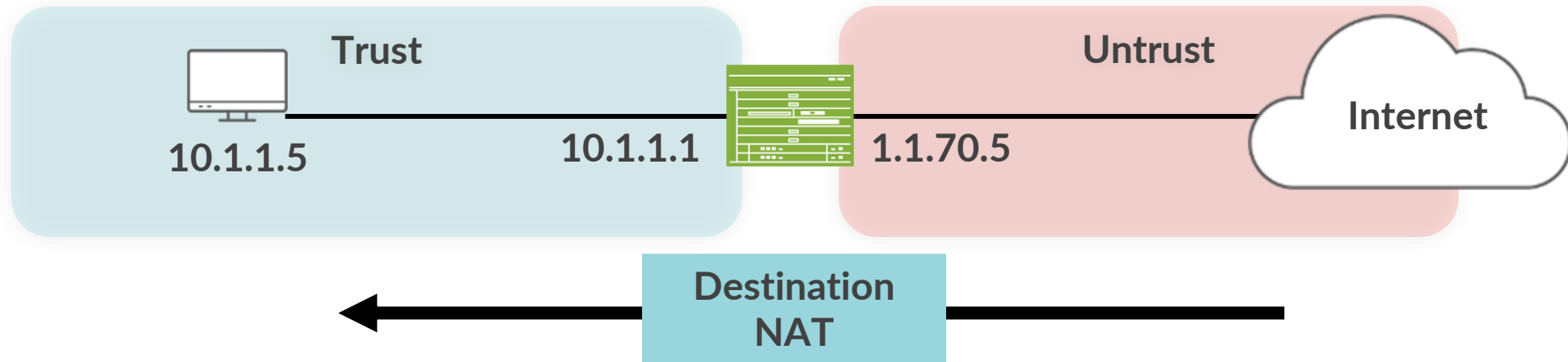
```

Pool Name	Address Range	Routing Instance	PAT	Total Address
src_nat_pool_napt	100.100.100.20-100.100.100.29	default	yes	10

```
Total rules: 1
Rule name      Rule set      From      To      Action
napt_1        src_nat_napt trust      untrust src_nat_pool_napt
```

Destination NAT 概要

- 宛先 IP アドレスを変換
 - オプションで宛先ポート番号の変換 (PAT)
- Destination NAT
 - Pool-based NAT のみ対応
 - 1 : 1 マッピング
 - 1 : N マッピング (ポート変換による振り分け)



Destination NAT (1:1) 設定

- アドレスプールの設定

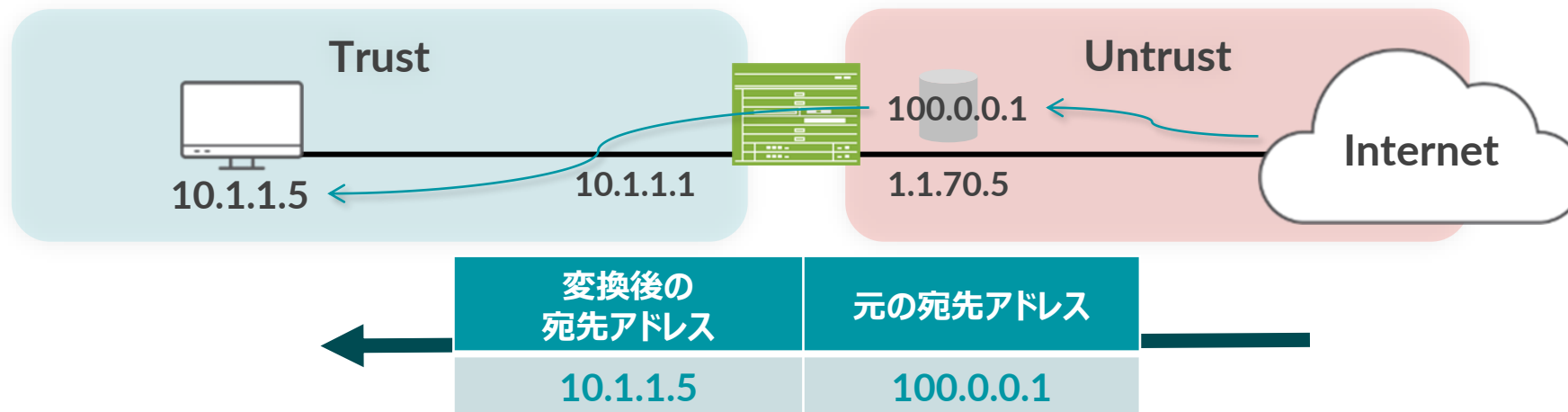
```
set security nat destination pool SVR_A address 10.1.1.5/32
```

- NAT ルールセットの設定

```
set security nat destination rule-set 1 from zone untrust
```

- NAT ルールの設定

```
set security nat destination rule-set 1 rule 1A match destination-address 100.0.0.1/32  
set security nat destination rule-set 1 rule 1A then destination-nat pool SVR_A
```



Destination NAT (1 : N) 設定

- アドレスプールの設定

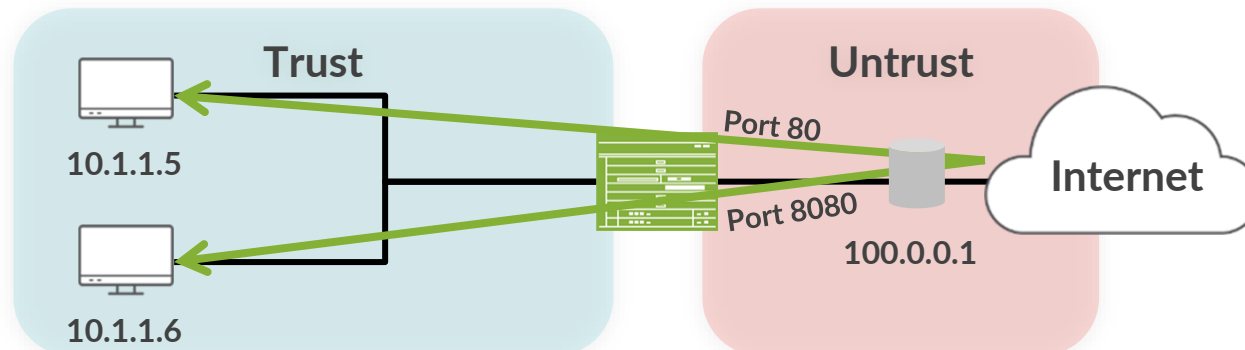
```
set security nat destination pool SVR_A address 10.1.1.5/32 port 80
set security nat destination pool SVR_B address 10.1.1.6/32 port 80
```

- NAT ルールセットの設定

```
set security nat destination rule-set 1 from zone untrust
```

- NAT ルールの設定

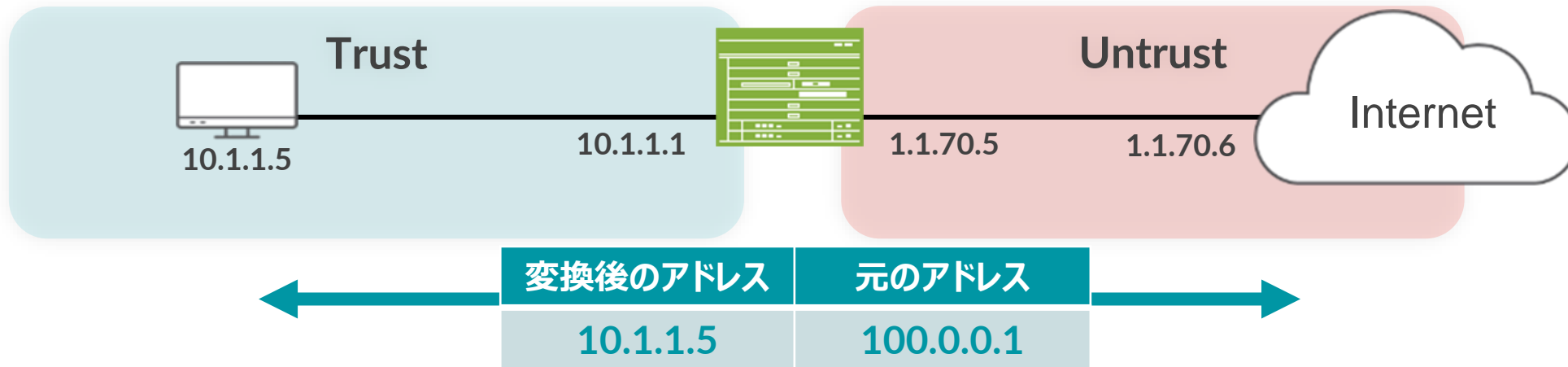
```
set security nat destination rule-set 1 rule 1A match destination-address 100.0.0.1/32
set security nat destination rule-set 1 rule 1A match destination-port 80
set security nat destination rule-set 1 rule 1A then destination-nat pool SVR_A
set security nat destination rule-set 1 rule 1B match destination-address 100.0.0.1/32
set security nat destination rule-set 1 rule 1B match destination-port 8080
set security nat destination rule-set 1 rule 1B then destination-nat pool SVR_B
```



変換後の宛先アドレス	元の宛先アドレス
10.1.1.5 Port 80	100.0.0.1 Port 80
10.1.1.6 Port 80	100.0.0.1 Port 8080

Static NAT 概要

- 1 : 1 でアドレスをマッピングして変換
 - ポート変換動作はなし
 - 双方向に通信を開始可能



Static NAT の設定

```
set security nat static rule-set R1 from zone untrust
set security nat static rule-set R1 rule 1A match destination-address 100.0.0.1/32
set security nat static rule-set R1 rule 1A then static-nat prefix 10.1.1.5/32
```

Static NAT 確認

- Untrust から 100.0.0.1 に対して Ping を実行


```
user@SRX> show security flow session
Session ID: 1367, Policy name: 2/5, Timeout: 2, Valid
  In: 1.1.70.6/256 --> 100.0.0.1/0;icmp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 1, Bytes: 84,
  Out: 10.1.1.5/0 --> 1.1.70.6/256;icmp, Conn Tag: 0x0, If: ge-0/0/2.0, Pkts: 1, Bytes: 84,
```

- 10.1.1.5 から Untrust に対して Ping を実行
 - 逆方向の Static Source NAT は自動的に有効化される

```
user@SRX> show security flow session
Session ID: 739, Policy name: test/6, Timeout: 2, Valid
  In: 10.1.1.5/1 --> 1.1.70.6/105;icmp, Conn Tag: 0x0, If: ge-0/0/2.0, Pkts: 1, Bytes: 60,
  Out: 1.1.70.6/105 --> 100.0.0.1/1;icmp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 1, Bytes: 60,
```

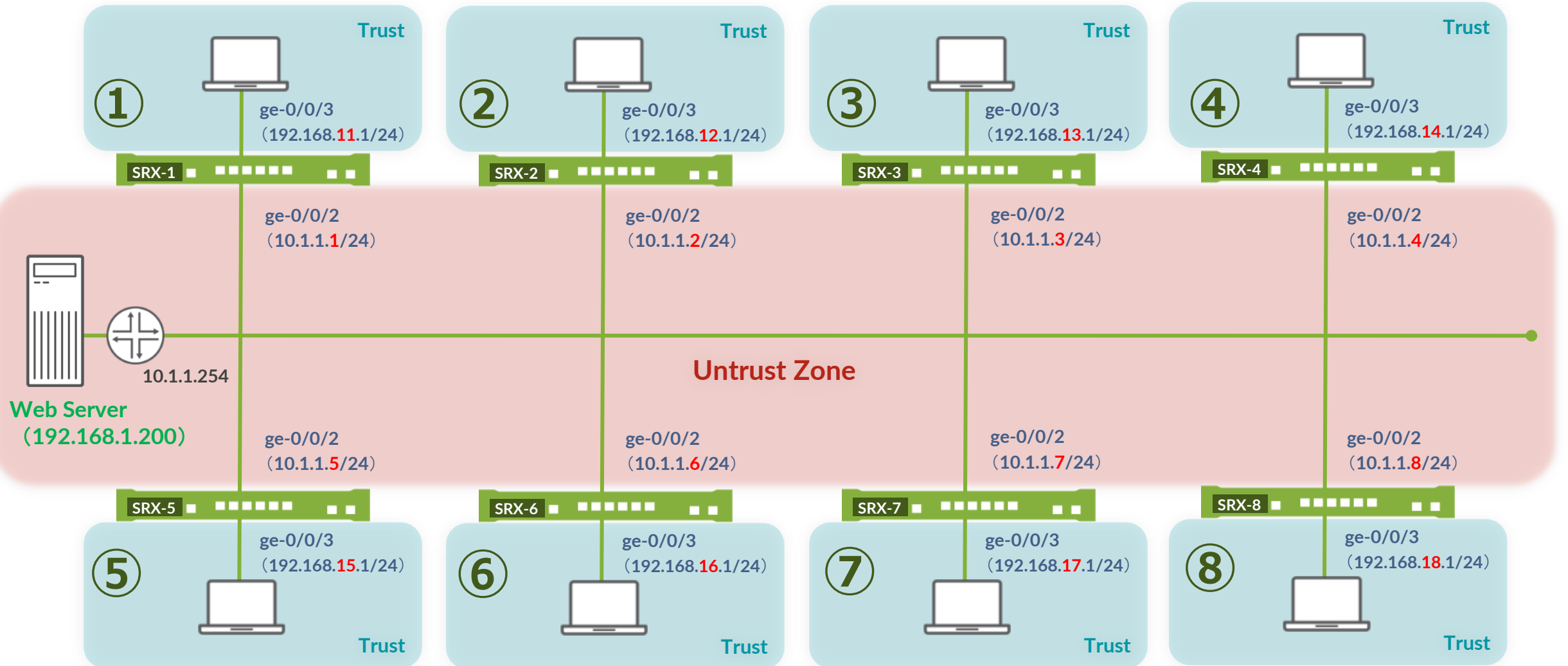
NAT 設定 & 動作確認コマンド

- セッション情報の確認
 - `show security flow session`
- Source NAT の確認
 - `show security nat source pool <pool-name | all>`
 - `show security nat source rule <rule-name | all>`
 - `show security nat source summary`
- Destination NAT の確認
 - `show security nat destination pool <pool-name | all>`
 - `show security nat destination rule <rule-name | all>`
 - `show security nat destination summary`
- Static NAT の確認
 - `show security nat static rule <rule-name | all>`



LAB.3 NAT の設定

Security "SRX" Course Topology (Lab.3: NAT)



Interface-based Source NAT

- Trust 側の送信元 IP アドレスを、SRX の Untrust IP に変換

元の送信元 IP (Trust)	変換後の送信元 IP (Untrust)
192.168.1X.0/24	10.1.1.X

- IP アドレスを確認
 - ブラウザから 192.168.1.200 にアクセスし、表示される IP アドレスを確認

- ルールセットの設定

```
set security nat source rule-set interface-nat from zone trust
set security nat source rule-set interface-nat to zone untrust
```

- NAT ルールの設定

```
set security nat source rule-set interface-nat rule rule1 match source-address 0.0.0.0/0
set security nat source rule-set interface-nat rule rule1 then source-nat interface
commit
```

- 設定後、再度 IP アドレスを確認
 - ブラウザから 192.168.1.200 に再度アクセスし、表示される IP アドレスを確認

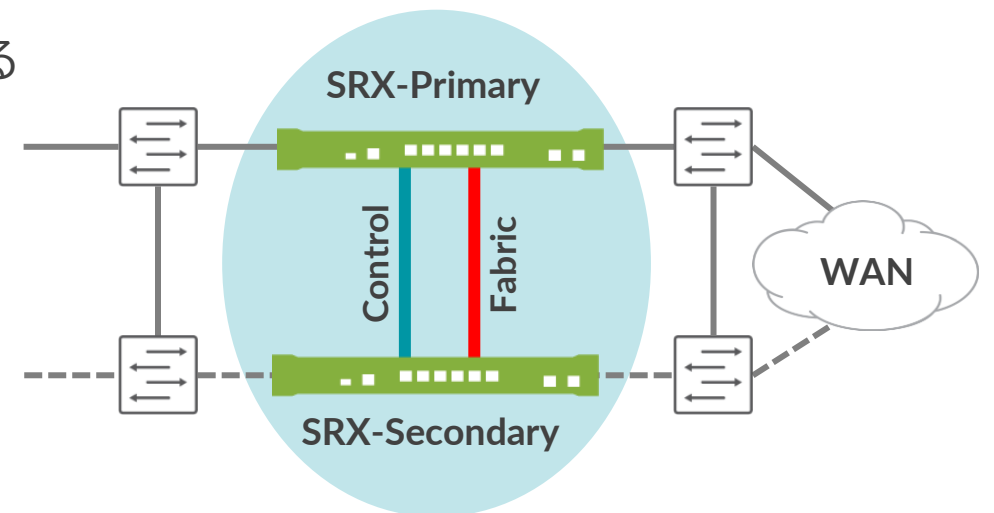
※結果例 ⇒ Backup Slides 資料参照



Chassis Cluster の設定

シャーシクラスタとは

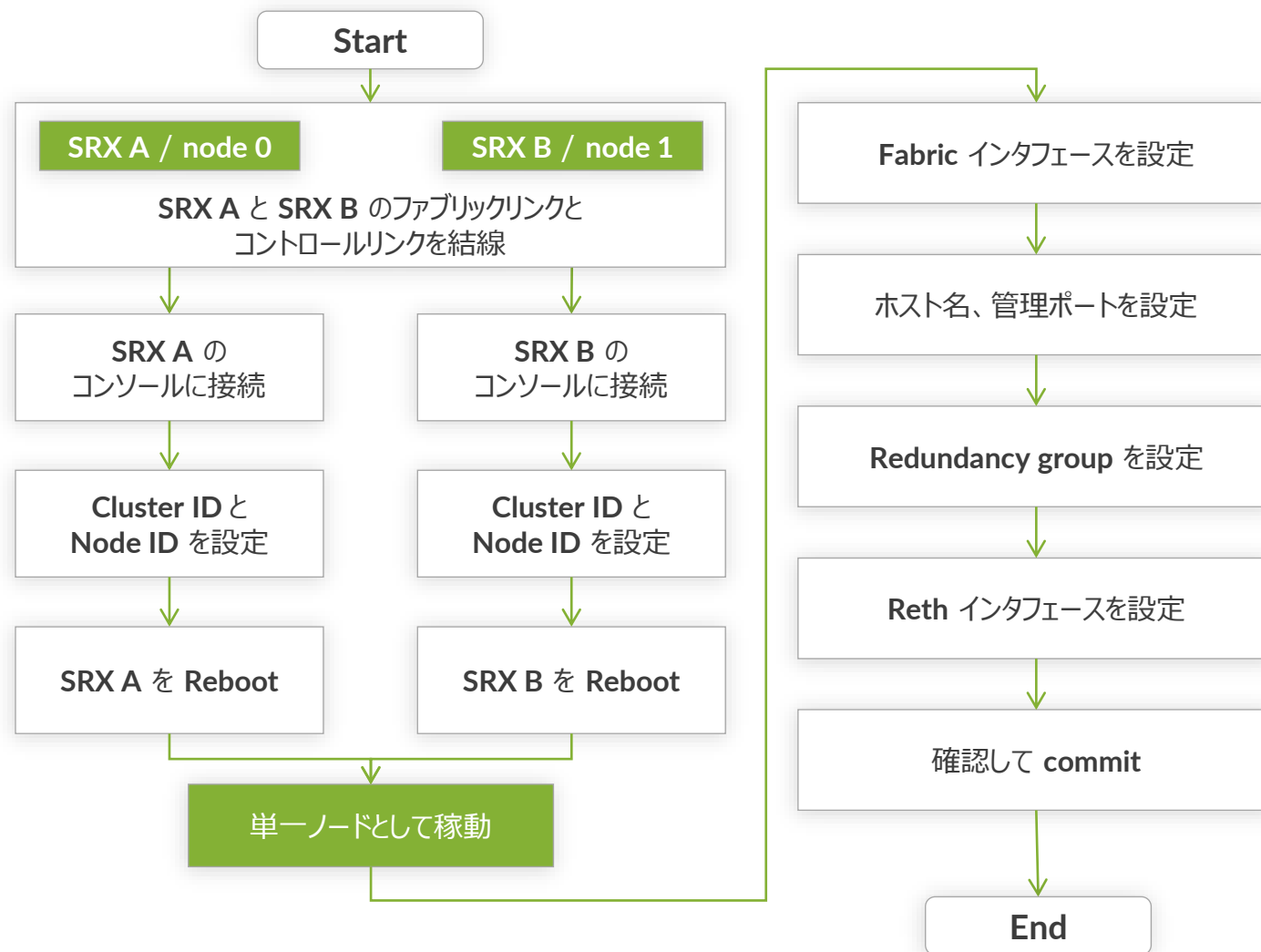
- 2 台の **SRX** シリーズを“単一ノード”として動作させるための機能
 - ステートフルフェールオーバーを実現
 - セッション情報や **Config** など、2 台で常に状態を同期
 - プライマリ機に障害が起きても、セカンダリ機が通信を継続
- シャーシクラスタの構成
 - 相互に状態同期と死活監視をするため、2 本の特別なリンクを設定
 - **コントロールリンク**
 - » コンフィグレーションとカーネルの状態を同期
 - » 機種ごとにどのポートが利用されるかが決まっている
 - **ファブリックリンク**
 - » セッション情報の同期とノード間のフロー処理
 - » 任意のポートに設定可能



シャーシクラスタの設定フロー

事前に確認

- ✓ 2 台の SRX が同じハードウェアであること
- ✓ 同じバージョンの OS であること
- ✓ 各種拡張セキュリティを使用時にはライセンスが同じ状態であること
*シャーシクラスタ用には不要



SRX300 シャーシクラスタポート構成

- **Control link (fxp1)** : node0 の ge-0/0/1 - node1 の ge-1/0/1
- **Fabric link (fab0/fab1)** : 任意のポート

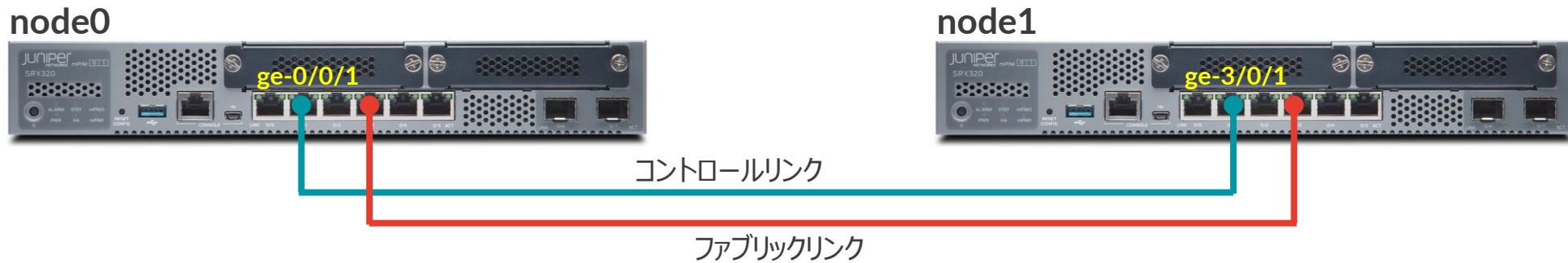


- インタフェーススロットのナンバリング



SRX320 シャーシクラスタポート構成

- **Control link (fxp1)** : node0 の ge-0/0/1 - node1 の ge-3/0/1
- **Fabric link (fab0/fab1)** : 任意のポート



- インタフェーススロットのナンバリング



SRX340 / 345 シャーシクラスタポート構成

- **Control link (fxp1)** : node0 の ge-0/0/1 - node1 の ge-5/0/1
- **Fabric link (fab0/fab1)** : 任意のポート

node0



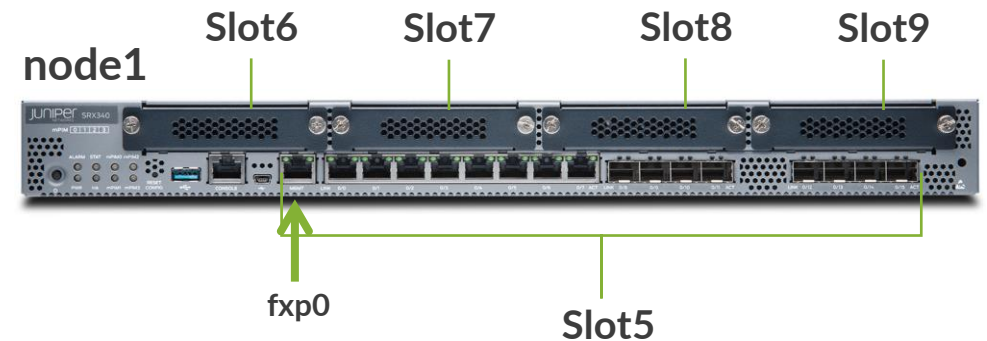
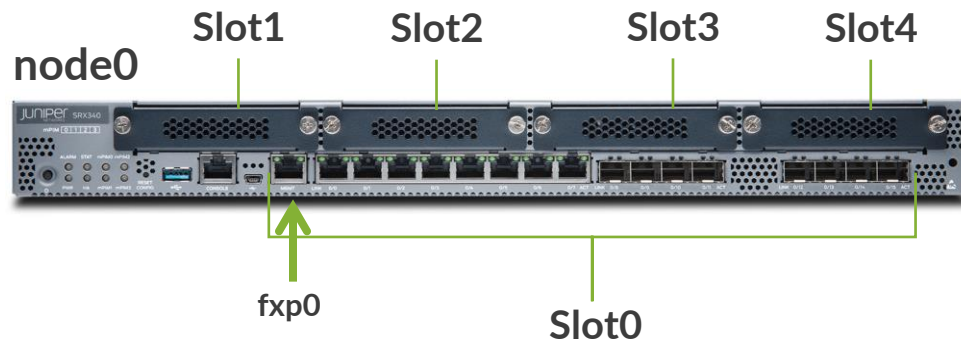
node1



コントロールリンク

ファブリックリンク

- インタフェーススロットのナンバリング



SRX380 シャーシクラスポート構成

- **Control link (fxp1)** : node0 の ge-0/0/1 - node1 の ge-5/0/1
- **Fabric link (fab0/fab1)** : 任意のポート

node0



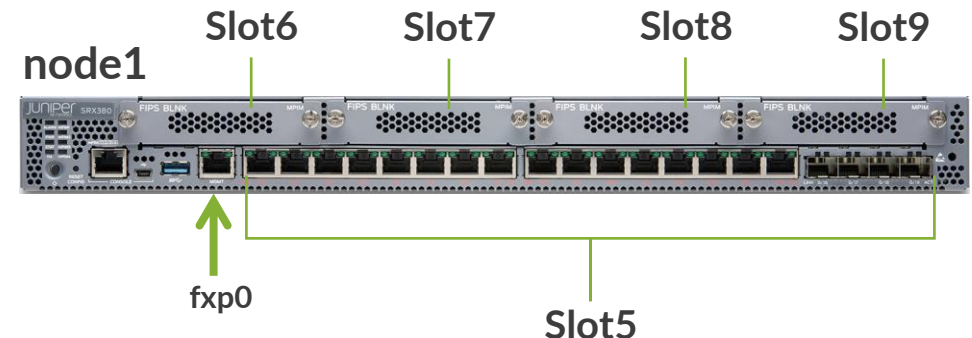
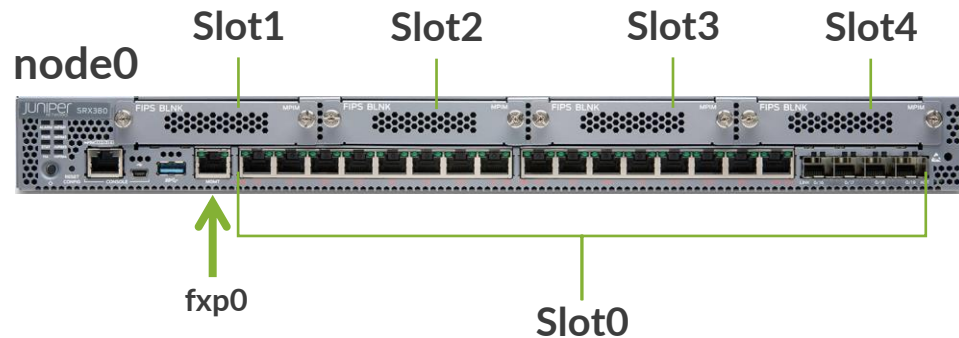
node1



コントロールリンク

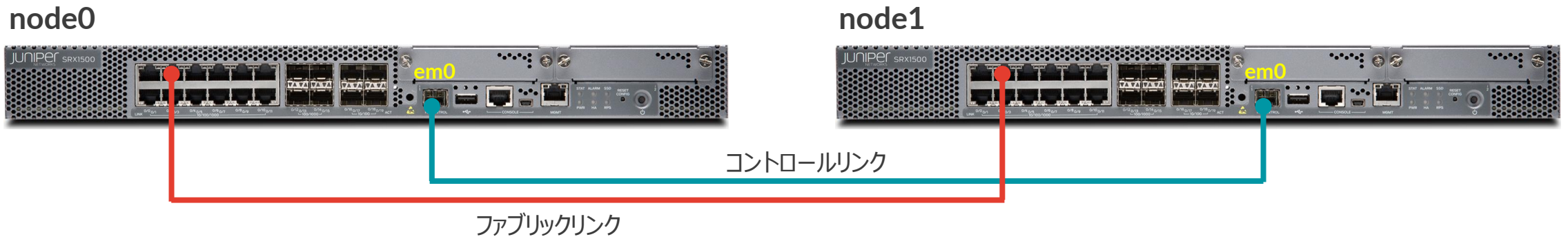
ファブリックリンク

- インタフェーススロットのナンバリング

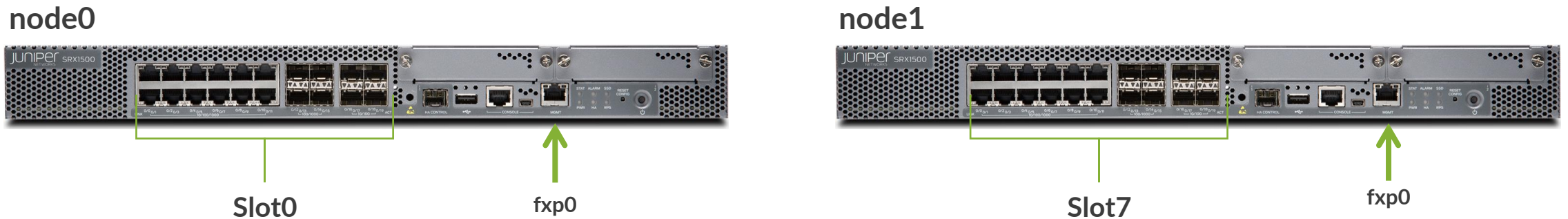


SRX1500 シャーシクラスタポート構成

- **Control link (em0)** : 専用コントロールポート
- **Fabric link (fab0/fab1)** : 任意のポート

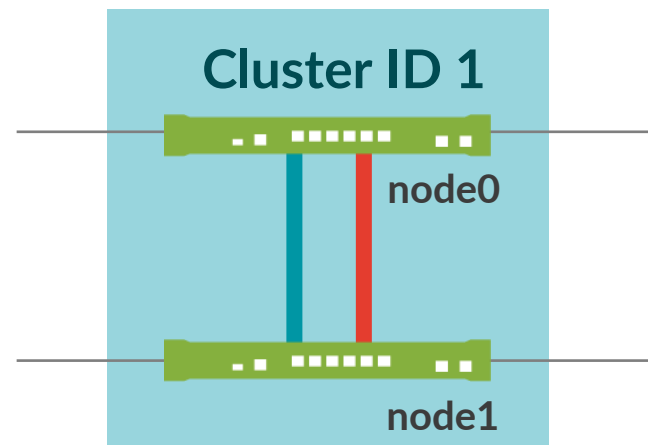


- インタフェーススロットのナンバリング



Cluster ID と Node ID

- **Cluster ID**
 - クラスタごとに固有に設定する ID
 - 同じ L2 ドメインで **1~255** まで設定可能
- **Node ID**
 - クラスタ内でのデバイス固有の ID
 - **node0** か **node1** を指定
 - Node ID によりインターフェース番号はリナンバリングされる
 - どちらがプライマリ、セカンダリになるかは ID とは別にプライオリティで設定



各ノードにホスト名と管理ポートを設定

- 各ノード固有のホスト名と管理ポートを設定
 - シャーシクラスタでは両ノードが同じ **Config** を共有
 - ノード固有の **Config** を設定したい場合に、**groups** オプションを利用
 - node0** 用のホスト名、管理ポートを設定

```
set groups node0 system host-name SRX_node0
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.0.101/24
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.0.100/24 master-only
```

- node1** 用のホスト名、管理ポートを設定

```
set groups node1 system host-name SRX_node1
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.0.102/24
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.0.100/24 master-only
```

- 2つのグループ設定を適用

```
set apply-groups "${node}"
```

各ノードの fxp0 に
固有の IP を設定

Primary にログインできる
共通の IP を設定

各ノードに Cluster ID と Node ID を設定

- Cluster ID と Node ID を設定
 - Operational モードで、以下コマンドを実行

- node0

```
root@SRX> set chassis cluster cluster-id 1 node 0 reboot
```

- node1

```
root@SRX> set chassis cluster cluster-id 1 node 1 reboot
```

- これらの情報は EPROM に保存される (Config には保存されない)
- 設定を反映させるためには **reboot** が必要
- **reboot** オプションを使うと、コマンド実行直後に **reboot**

```
root@SRX> set chassis cluster cluster-id 1 node 0 reboot  
Successfully enabled chassis cluster. Going to reboot now.
```

```
root@SRX>  
*** FINAL System shutdown message from root@SRX ***
```

```
System going down IMMEDIATELY
```

シャーシクラスタの状態確認

- 各ノードの再起動後、クラスタが形成される
 - show chassis cluster status** コマンドで状態を確認

```
root@SRX_node0> show chassis cluster status
Monitor Failure codes:
  CS  Cold Sync monitoring          FL  Fabric Connection monitoring
  GR  GRES monitoring              HW  Hardware monitoring
  IF  Interface monitoring         IP  IP monitoring
  LB  Loopback monitoring          MB  Mbuf monitoring
  NH  Nexthop monitoring           NP  NPC monitoring
  SP  SPU monitoring              SM  Schedule monitoring
  CF  Config Sync monitoring       RE  Relinquish monitoring
  IS  IRQ storm

Cluster ID: 1
Node   Priority Status          Preempt Manual   Monitor-failures
-----
Redundancy group: 0 , Failover count: 0
node0  1          primary             no    no    None
node1  1          secondary           no    no    None
```

- 各ノードのプロンプト上のステータスを確認

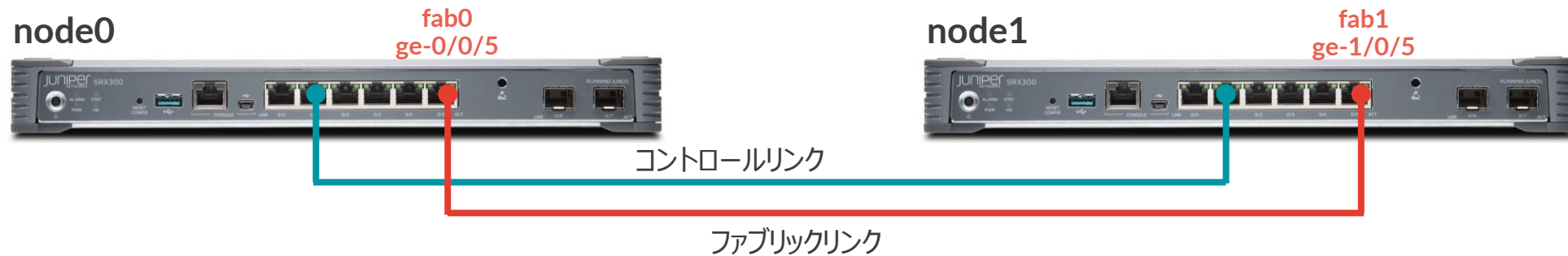
```
{primary:node0}
root@SRX_node0>
```

```
{secondary:node1}
root@SRX_node1>
```

ファブリックリンクを設定

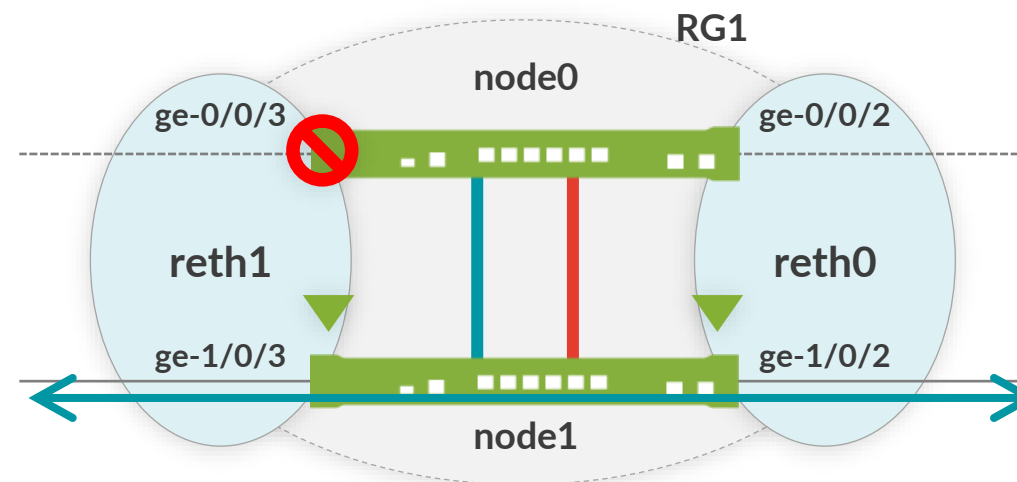
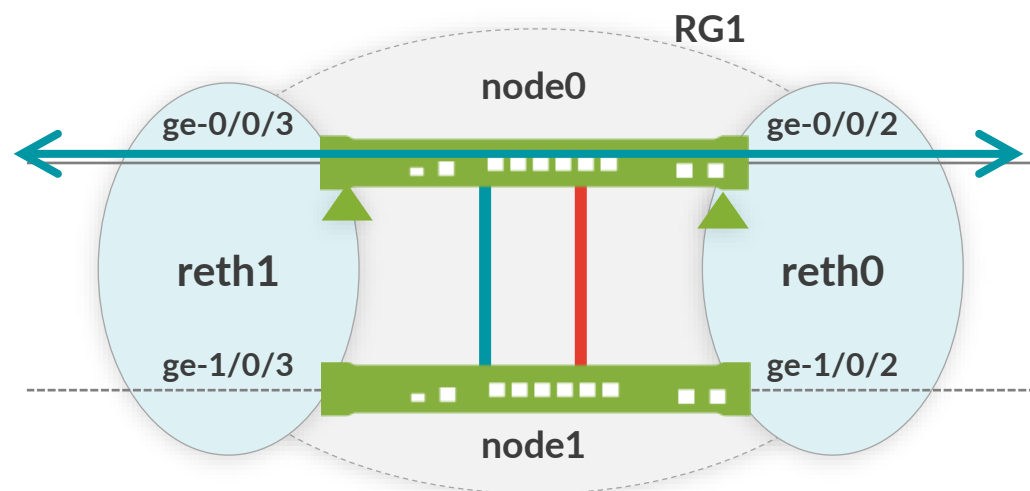
- ファブリックリンクを設定
 - **Configuration** モードで以下コマンドを設定
 - node0 側の ge-0/0/5 を fab0、node1 側の ge-1/0/5 を fab1 として設定

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/5
set interfaces fab1 fabric-options member-interfaces ge-1/0/5
```



Reth インタフェースと Redundancy Group ①

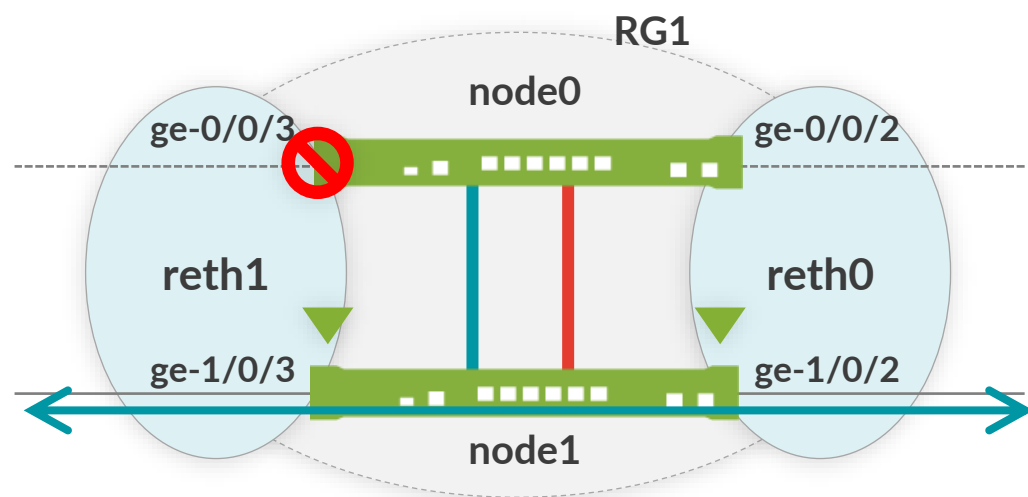
- **Redundant Ethernet Interface (reth)**
 - 2 台のノード間で共有される仮想のインタフェース
 - 各ノードの物理リンクを 1 つの reth にマッピング
 - どちらのノードが reth の転送を担当するか？
 - “Redundancy Group” でノードごとにプライオリティ付け
 - より高いプライオリティを持つノードがプライマリとして転送を担当
 - プライマリに障害が起きた場合
 - 同じ Redundancy Group に所属するすべての reth がセカンダリにフェールオーバー



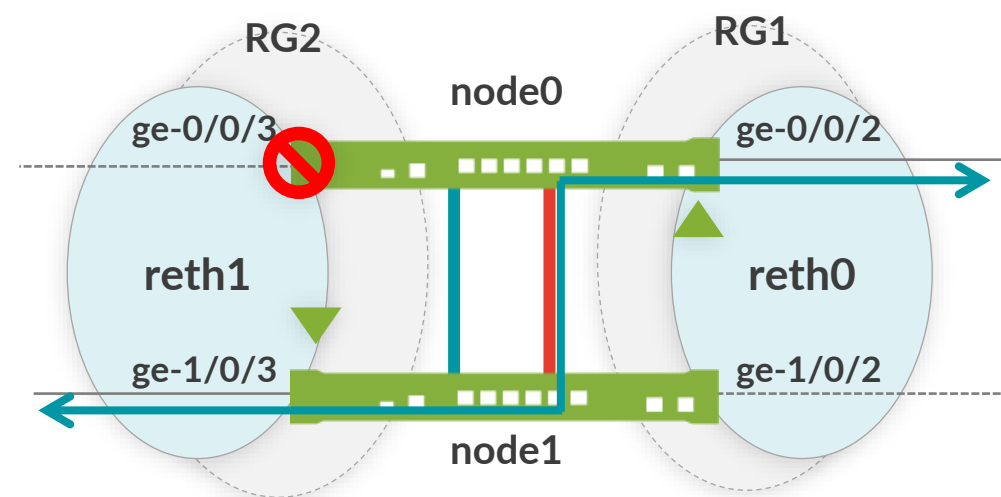
Reth インタフェースと Redundancy Group ②

- Redundancy Group (RG)

- 障害発生時にフェールオーバーの影響を共有する範囲を指定するためのグループ
 - 1つのグループ内では、どちらかのノードがプライマリとして処理を担当
 - グループごとにノードにプライオリティを設定し、プライマリノードを決定



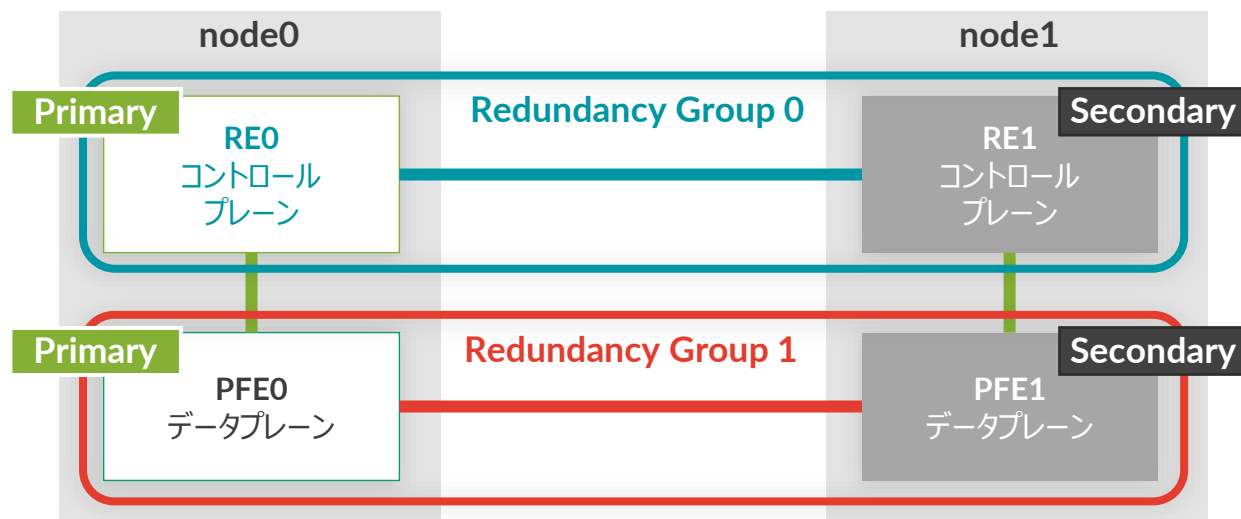
RG1 に reth0 と reth1 をバインドした場合
(障害時には reth0 と reth1 が Failover)



RG1 に reth0 と RG2 に reth1 をバインドした場合
(障害時には reth1 だけが Failover)

Reth インタフェースと Redundancy Group ③

- Redundancy Group (RG) 0 と 1+
 - RG0 は Chassis Cluster を制御するコントロールプレーン用の RG としてシステムに予約される
 - Redundancy Group 0
 - ルーティングエンジン (コントロールプレーン) の Redundancy Group
 - Redundancy Group 1~ 以上
 - reth インタフェース (データプレーン) の Redundancy Group

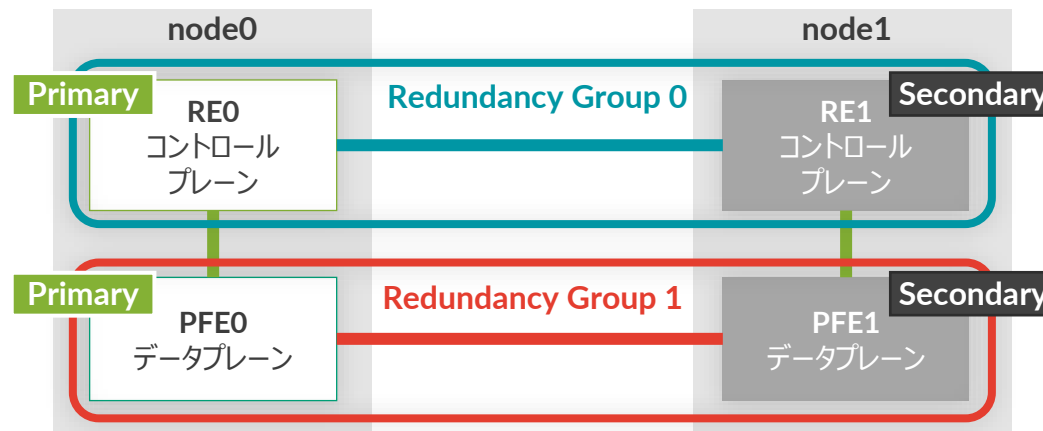


Redundancy Group の設定

- 各ノードを **Redundancy Group** に所属させ、プライオリティを設定
 - コンフィグレーションモードで以下コマンドを設定

```
set chassis cluster redundancy-group 0 node 0 priority 200
set chassis cluster redundancy-group 0 node 1 priority 100
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

- Redundancy Group 0
 - ルーティングエンジン (RE) 共有のグループ
- Redundancy Group 1 ~ 以上
 - インタフェース (PFE) 共有のグループ



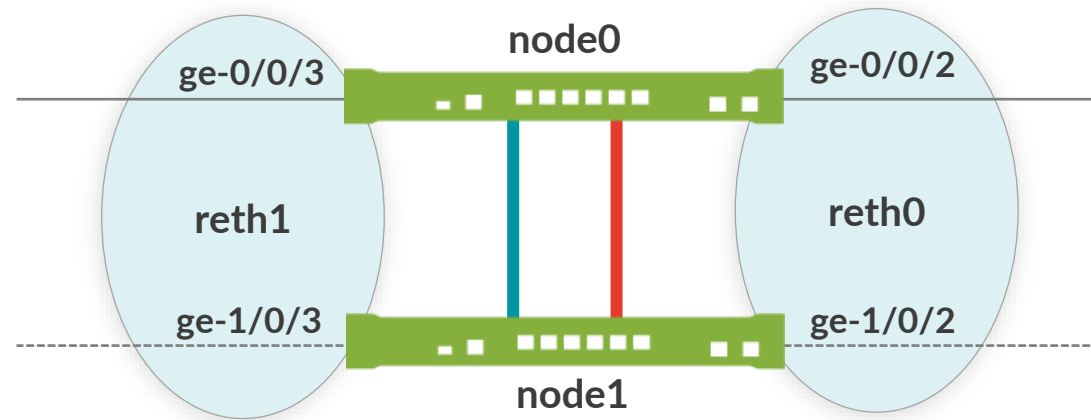
Redundant Ethernet Interface (reth) の設定

- コンフィグレーションモードで以下コマンドを設定
 - クラスタ内の reth インタフェースの総数を定義

```
set chassis cluster reth-count 2
```

- reth にバインドする物理（または論理）インタフェースを設定

```
set interfaces ge-0/0/2 gigether-options redundant-parent reth0
set interfaces ge-1/0/2 gigether-options redundant-parent reth0
set interfaces ge-0/0/3 gigether-options redundant-parent reth1
set interfaces ge-1/0/3 gigether-options redundant-parent reth1
```



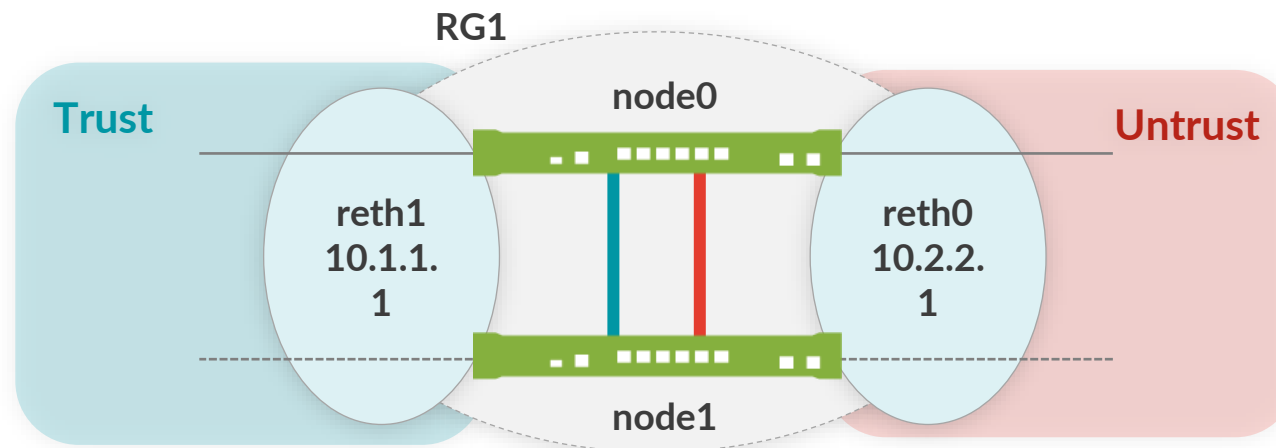
Redundant Ethernet Interface (reth) の設定

- reth を Redundancy Group に所属させ、IP アドレスを設定

```
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.1.1.1/24
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.2.2.1/24
```

- reth を Security Zone にバインド
 - reth1 を trust に、reth0 を untrust にバインド

```
set security zones security-zone trust interfaces reth1.0
set security zones security-zone untrust interfaces reth0.0
```



プリエンプトとインタフェースモニタリングの設定

- プリエンプトとインタフェースモニタリングの設定
 - **Redundancy Group** にプリエンプトを設定
 - 障害復旧時にプライオリティの高いノード側をプライマリに戻す動作

```
set chassis cluster redundancy-group 1 preempt
```

- インタフェースモニタリングを設定
 - **RG** ごとに切り替わりのトリガーとなるインタフェースと **weight** を指定

```
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/3 weight 255
```

- **RG** ごとに **255** の値を持つ
- インタフェースがダウンすると **255** から **weight** に設定した数値が引かれる
- 値が **0** 以下になるとフェールオーバー

シャーシクラススタ確認コマンド ①

- **show interfaces terse** : インタフェースの確認
 - ファブリックリンクの確認

```
root@SRX_node0> show interfaces terse | match fab
ge-0/0/5.0          up    up    aenet  --> fab0.0
ge-1/0/5.0          up    up    aenet  --> fab1.0
fab0                up    up
fab0.0              up    up    inet   30.17.0.200/24
fab1                up    up
fab1.0              up    up    inet   30.18.0.200/24
```

- **reth** インタフェースの確認

```
root@SRX_node0> show interfaces terse | match reth
ge-0/0/2.0          up    up    aenet  --> reth0.0
ge-0/0/3.0          up    up    aenet  --> reth1.0
ge-1/0/2.0          up    up    aenet  --> reth0.0
ge-1/0/3.0          up    up    aenet  --> reth1.0
reth0               up    up
reth0.0             up    up    inet   10.2.2.1/24
reth1               up    up
reth1.0             up    up    inet   10.1.1.1/24
```

シャーシクラスタ確認コマンド ②

- **show chassis cluster interface**

- クラスタに所属するインタフェースの確認

```
root@SRX_node0> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index   Interface   Monitored-Status   Internal-SA   Security
  0       fxp1       Up                 Disabled      Disabled

Fabric link status: Up

Fabric interfaces:
  Name     Child-interface   Status                               Security
                               (Physical/Monitored)
  fab0    ge-0/0/5          Up / Up                             Disabled
  fab0
  fab1    ge-1/0/5          Up / Up                             Disabled
  fab1

Redundant-ethernet Information:
  Name     Status   Redundancy-group
  reth0    Up       1
~~~~~
```


シャーシクラスタ確認コマンド ③

- **show chassis cluster status**

- シャーシクラスタのステータスを表示

```
root@SRX_node0> show chassis cluster status
Monitor Failure codes:
  CS  Cold Sync monitoring          FL  Fabric Connection monitoring
  GR  GRES monitoring                HW  Hardware monitoring
  IF  Interface monitoring           IP  IP monitoring
  LB  Loopback monitoring            MB  Mbuf monitoring
  NH  Nexthop monitoring             NP  NPC monitoring
  SP  SPU monitoring                 SM  Schedule monitoring
  CF  Config Sync monitoring         RE  Relinquish monitoring
  IS  IRQ storm

Cluster ID: 1
Node   Priority Status          Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 0
node0  200     primary          no    no    None
node1  100     secondary        no    no    None

Redundancy group: 1 , Failover count: 1
node0  200     primary          yes   no    None
node1  100     secondary        yes   no    None
```

シャーシクラスタ確認コマンド ④

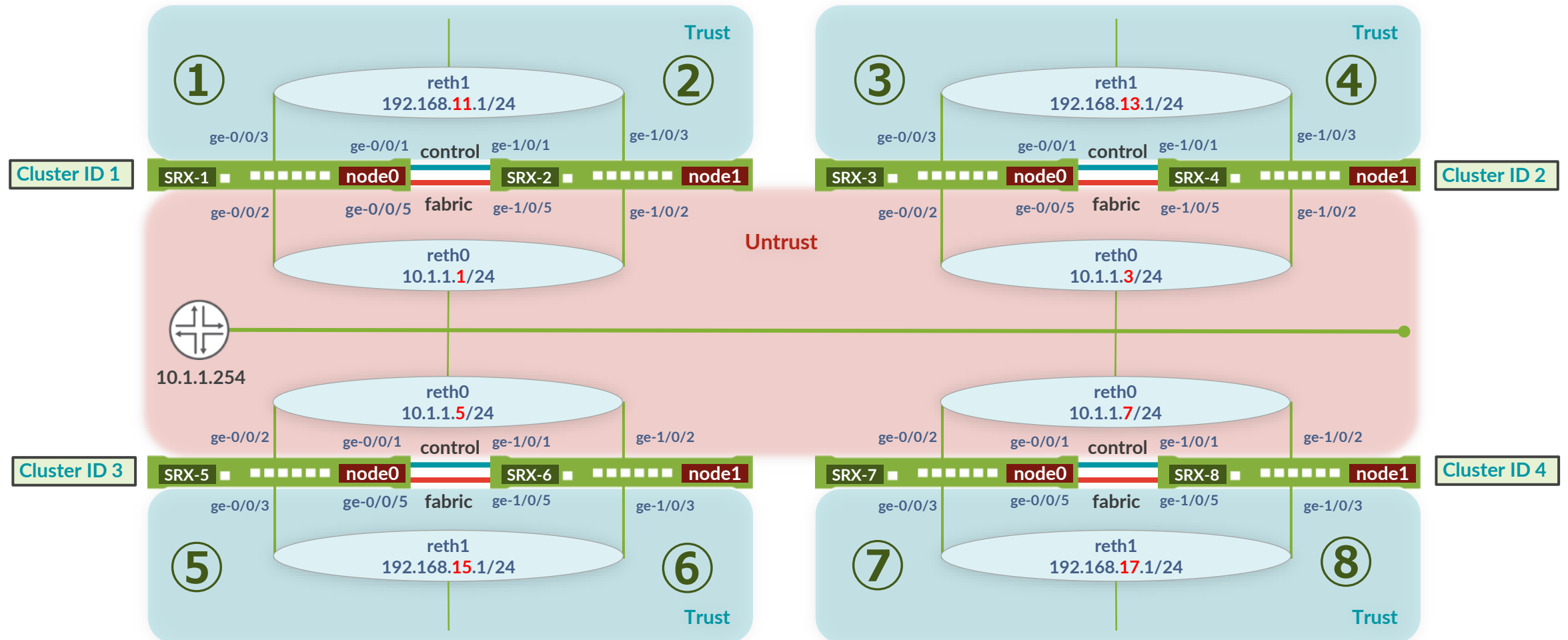
- **show chassis cluster statistics**
 - シャーシクラスタの統計情報

```
root@SRX_node0> show chassis cluster statistics
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 3536
    Heartbeat packets received: 3140
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 4527
    Probes received: 4526
  Child link 1
    Probes sent: 0
    Probes received: 0
Services Synchronized:
  Service name          RTOs sent  RTOs received
  Translation context   0          0
  Incoming NAT          0          0
~~~~~
```



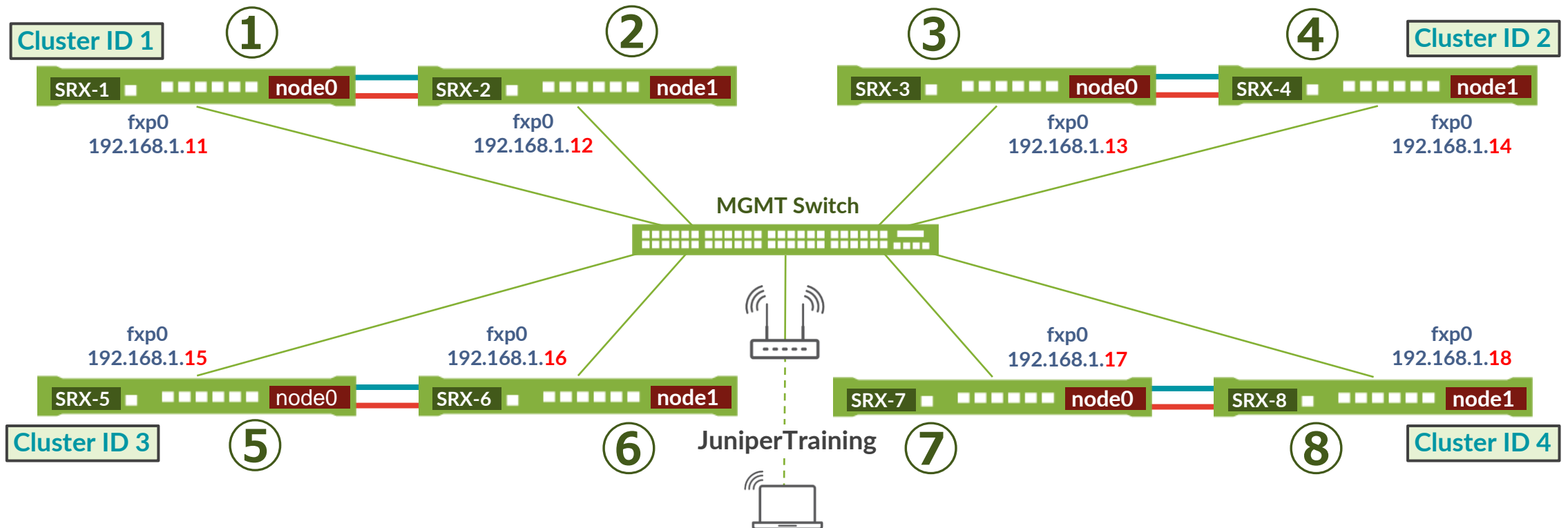
LAB.4 Chassis Cluster の設定

Security "SRX" Course Topology (Lab.4: Chassis Cluster)

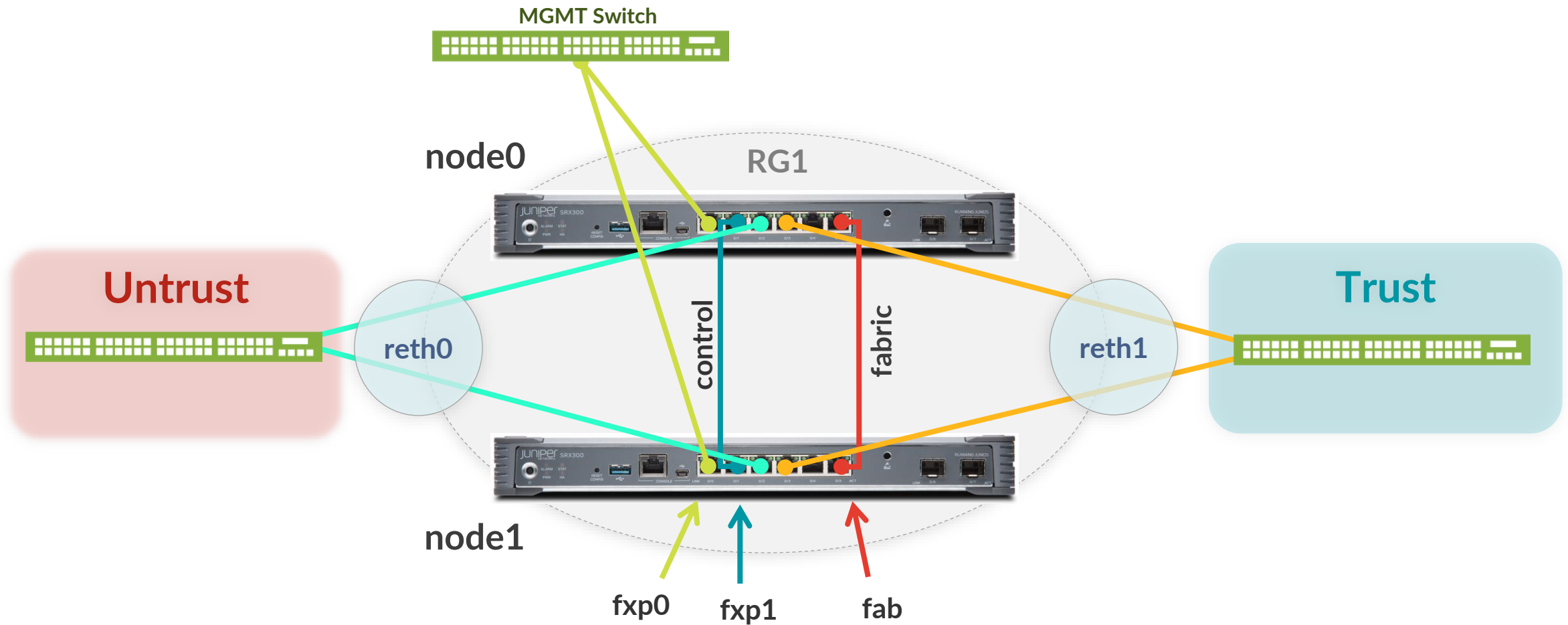


Security "SRX" Course Topology (Lab.4: Chassis Cluster Management)

- 管理用 IP アドレス一覧
 - fxp0 に設定する IP アドレス



SRX300 シャーシクラスポート構成



Chassis Cluster の設定

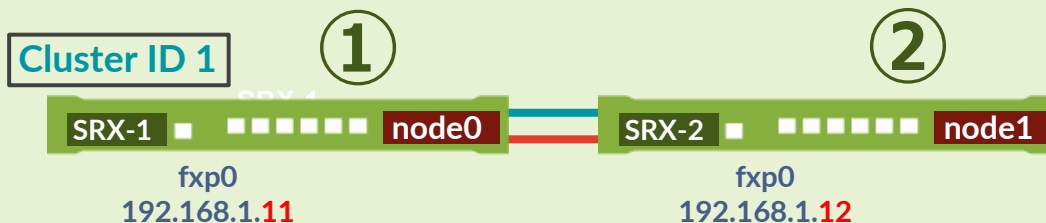
- トポロジー図に従って **Chassis Cluster** を組む
 - **Active / Passive** 構成
 - コントロールリンクとファブリックリンクは **1** 本ずつ用意
 - **RETH0** を **Untrust** 側インタフェースとして構成
 - **RETH1** を **Trust** 側インタフェースとして構成
- 以下のコマンドで、ステータスを確認
 - **show chassis cluster status**
 - **show chassis cluster interface**
 - **show chassis cluster statistics**

Cluster ID と Node ID の設定

- Node 固有の設定を追加（座席番号が奇数の方のみ）
- IP アドレスは管理用 IP アドレス図を参照

```
set groups node0 system host-name SRX-x_node0
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.1.1x/24
set groups node1 system host-name SRX-y_node1
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.1.1y/24
set apply-groups "${node}"
commit
```

Chassis Cluster の構成例



- ① node0
host-name : SRX-1_node0
address : 192.168.1.11/24
- ② node1
host-name : SRX-2_node1
address : 192.168.1.12/24

Cluster ID と Node ID の設定

- コントロールリンクとファブリックリンクを結線後（※結線済み）、以下を実行
- **node0** で実行後、3 秒以上あけてから **node1** で実行

- **node0**（座席番号が奇数）

```
lab@SRX> set chassis cluster cluster-id X node 0 reboot
```

- **node1**（座席番号が偶数） ※ 3 秒以上後に

```
lab@SRX> set chassis cluster cluster-id X node 1 reboot
```

- コマンド実行後、即時 **reboot** に入りセッションが切断される

再起動後の状態確認

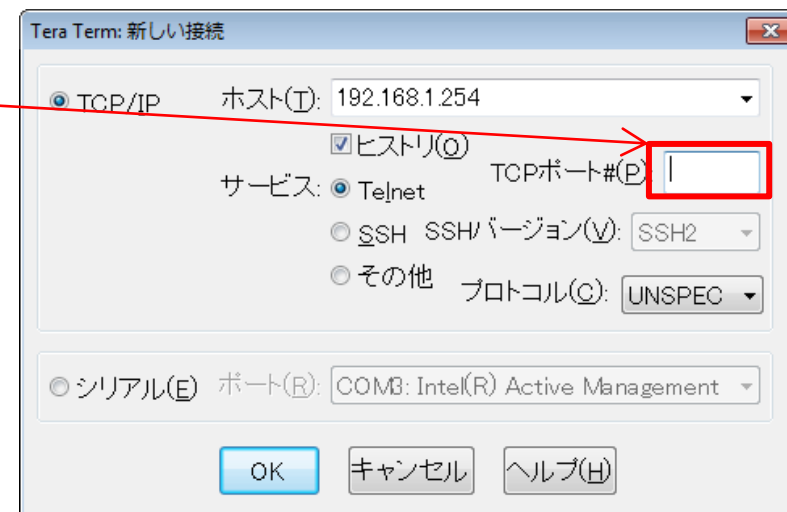
- **Wireless の SSID を “JuniperTraining” に変更**
 - PC の IP アドレスが 192.168.1.xx に変わったことを確認
 - Tera Term を立ち上げ、192.168.1.254 に telnet
 - **TCP ポートは [2008+座席番号]**
 - 座席番号 ① 2008+1 = 2009
 - 座席番号 ② 2008+2 = 2010
 - ...

- 起動後、単一ノードとして稼動状態を確認
 - 設定した管理用 IP アドレスに telnet でログイン
 - CC が組めていれば以下のようにステータスが表示される

```
{primary:node0}  
lab@SRX-1>
```

```
{secondary:node1}  
lab@SRX-1>
```

- show chassis cluster status コマンドでステータスを確認
 - Redundancy Group 0 のステータスが以下のようにになっていること
 - node0 Primary
 - node1 Secondary



ファブリックリンクの設定

- 以下の設定変更はすべて **Primary (node0)** から実施

① これまでの **Lab** で設定した不要な **Config** を削除

```
delete system host-name
delete system services dhcp-local-server
delete interfaces
delete security nat
delete security zones security-zone trust interfaces
delete security zones security-zone untrust interfaces
```

② **Fabric** リンクの設定を追加

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/5
set interfaces fab1 fabric-options member-interfaces ge-1/0/5
```

Redundant Group (RG) と Reth インタフェースの設定

- 以下の設定変更はすべて **Primary (node0)** から実施
 - **RG0 と RG1** を設定

```
set chassis cluster redundancy-group 0 node 0 priority 200
set chassis cluster redundancy-group 0 node 1 priority 100
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

- **reth0 と reth1** を設定

```
set chassis cluster reth-count 2
set interfaces ge-0/0/2 ether-options redundant-parent reth0
set interfaces ge-1/0/2 ether-options redundant-parent reth0
set interfaces ge-0/0/3 ether-options redundant-parent reth1
set interfaces ge-1/0/3 ether-options redundant-parent reth1

set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.x/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 192.168.1x.1/24
```

インタフェースモニタリングとプリエンプトの設定

- 以下の設定変更はすべて **Primary (node0)** から実施
 - **reth** を **Security Zone** にバインド

```
set security zones security-zone trust interfaces reth1.0
set security zones security-zone untrust interfaces reth0.0
```

- インタフェースモニタリングとプリエンプトを設定

```
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/3 weight 255
set chassis cluster redundancy-group 1 preempt
```

```
commit
```

Chassis Cluster の動作確認

- 以下のコマンドで、ステータスを確認
 - `show chassis cluster status`
 - `show chassis cluster interface`
 - `show chassis cluster statistics`

※結果例 ⇒ Backup Slides 資料参照

- 障害動作確認
 - SRX から **10.1.1.254** に対して **Ping** を実行
 - **reth0** のプライマリリンク (**node0** の **ge-0/0/2**) のケーブルを抜線
 - **Ping** 通信が継続していることを確認

- 抜いたケーブルを元に戻す



THANK YOU

JUNIPER
NETWORKS

Driven by
Experience™



Appendix

- A. TIPs to be Junos Experts
- B. Chassis Cluster Deep Dive
- C. IPsec VPN の設定
- D. NAT Pool Options
- E. Security Logging
- F. Firewall Filter (ACL) の設定



Appendix A: TIPs to be Junos Experts

俳句の表示

- 検証作業やトラブルシュー트에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
    }
  }
  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 {
```



```
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、インタフェー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、複数のインタフェー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** 内の個別インタフェー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


```
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** インタフェースの **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;
  }
}
```

インタフェース名やエリア名、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

特定のインタフェー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: Chassis Cluster Deep Dive

Cluster と Node ID

- **Cluster ID**

- シャーシ間でクラスタリングの設定をする際に、**Cluster ID** が必要
- **Cluster ID** は、**1** から **255** まで、割り振ることが可能
- 注意点としては、同じレイヤ **2** ブroadcastキャストセグメントで他の **Cluster ID** と重複しないようにしなければならない

- **Node ID**

- **Cluster** 内で各々のメンバーは、**Node ID** (**0** または **1**) により識別される
- 現在サポートされているノード数は、最大 **2** 台
- **Node ID** と **Cluster ID** は、**EPROM** に、保存される
- コンフィギュレーションを初期設定に戻しても、**Cluster** の **Disable** を実施しないと **Cluster** は解除されない

ノード独自（固有）のコンフィグ

- ノード固有のコンフィグ
 - Junos では、両機器に、同じコンフィグレーションを保持しつつ従ってコンフィグは、**Primary** 側で実施
 - コンフィグの独自区分は、ノード番号（**EPROM** に保存）により示される
 - どのノードがどのグループ所属するなどを定義するためには、**Junos** グループ機能を利用
 - ノード固有のコンフィグには以下が含まれる
 - **fxp0** のコンフィグ： マネージメントポート
 - システム名（ホストネーム）
 - バックアップルータ **IP** アドレス

コントロールポート（コントロールリンク）

- コントロールポート（コントロールリンク）
 - コントロールポートは、RE 間のコミュニケーションを許可
 - **Cluster** メンバー間で、**JSRP**、**chassisd**、カーネルの情報を共有
 - 現在、各々の機器に割り当てることができるコントロールポートは、ひとつだけです。（**fxp1**）が割り当てられる
 - **SRX** ブランチシリーズは、コントロールポートが自動的に割り振られるため、**コンフィグをする必要がない**

ファブリックポート（ファブリックリンク）

- ファブリックポート（ファブリックリンク）
 - データプレーンを直接つなぐファブリックポート
 - **Cluster** メンバー間で、同一のデータプレーンを接続
 - **Cluster** 全体でサポートされているファブリックリンクは、最大 2 リンク
 - **SRX HA** にて、**RTO** メッセージは、ファブリックリンク（セッション、ルートなど）を介して同期
 - **Active / Active** 構成では、データは、メンバー間のファブリックポートを介して（**Z** 型）通信可能
 - 非対称のデータ（ユーザー）トラフィックもサポートされる
- ファブリックポート（ファブリックリンク） コンポーネント
 - **fab0** と **fab1** の仮想インタフェースは、**node0** と **node1** をつなぐために、作成することが必要
 - **node0** 側に **fab0** インタフェースを作成し、**node1** 側に **fab1** インタフェースを作成し、直接結線することが推奨される

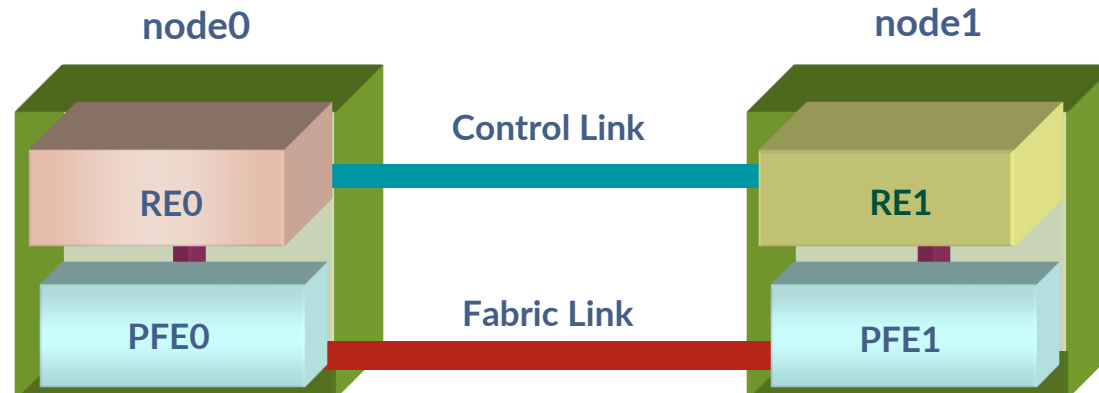
コントロールポートとファブリックポートの注意事項

- コントロールポートとファブリックポートにスイッチを挟む場合
 - コントロールリンクとファブリックリンクの **VLAN** を分けることが必要
 - 遅延は、**100msec** 以下にしてください
 - **IGMP Snooping** 機能は、無効にしてください
 - コントロールリンクとファブリックリンクの **VLAN** に他のトラフィックを流さないことが必要
 - トラフィックを、カプセルングする際は、**MTU** のサイズに注意が必要
 - パケットのフラグメントをサポートされない

Redundancy Group

- **Redundancy Group**

- コンポーネントをグループ化し、シャーシ間をフェイルオーバーする
- **Redundancy Group 0** は、ルーティングエンジンとして使われる
- **Redundancy Group 1** は、**Active / Passive** の **Redundant interface** として使われる
Redundancy Group 1 以上は、**Active / Active** の時に使われる
- オペレーションは、**ScreenOS** の **VSD** に非常によく似ています。**Junos** では、コントロールプレーンとデータプレーンを分けるために、少なくともふたつの **Redundancy Group** が必要
- **Redundancy Group 0** は、コントロールプレーン冗長の為に、**Redundancy Group 0** にマッピングされ、**Redundancy Group 1** 以上は、データプレーンにマッピングされる



Redundant Ethernet Interface の設定

• Redundant Interface

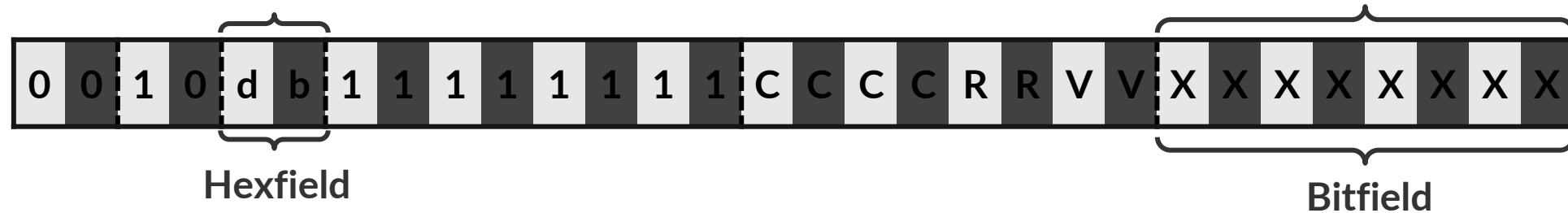
- Redundant Interface は、Active / Passive としての役割を持つメンバーインタフェースを構成する仮想インタフェース
 - SRX の Active / Active とは、各々の Redundant Ethernet メンバーが Active / Active になるわけではなく、異なる Redundancy Group を利用して、同時にトラフィックを転送できる構成または、状態を示す（それぞれの Redundancy Group の Master をイレコにする）
- シャーシ跨ぎのトラフィックの概念を除いて ScreenOS と Redundant Interface の考え方は同じ
- コンフィグでは、`reth <番号 X >` となる
- すべてのロジカルコンフィグは、このインタフェースにすることが必要
- 物理インタフェースとは、異なる
- 例えば、IP アドレス、QoS、Zone、VPN などの設定に相当
- 物理プロパティだけは、メンバーインタフェースに適応される

• Redundant Interface の作成

- リンクアグリゲーションインタフェースを作成するように、作成することが可能
- SRX が仮想インタフェースを作成するために、シャーシ内で `reth` 番号を割り振る必要がある
- `reth interface` を作成したら、`reth interface` を Redundancy Group にバインドする必要がある

Redundant Interface MAC アドレス

- Cluster ID を利用して、Reth MAC アドレスが提供される
 - Reth MAC アドレスの構成



- 構成要素：
 - CCCC - Cluster ID、ユーザにより割り振られた ID 番号
 - RR - Reserved, 00.
 - VV - Version、ファーストリリースは、00
 - XXXXXX - Interface ID、Reth Index から決定される

- Cluster ID 1、Reth Interface 0のMACアドレスのフォーマット例：



インタフェース モニタリング

- インタフェース モニタリング
 - **Cluster** 内のリンクダウンやインタフェースのリアクションのモニター機能
 - **ScreenOS** のように、閾値（**255**）からウェイトの値にて減算利用し、シャーシ内でのフェイルオーバーを実現
 - リモートの障害とフェイルオーバーを関連付けるためには、**Junos 11.2** 以降でサポートされている **IP Monitoring** の機能が必要

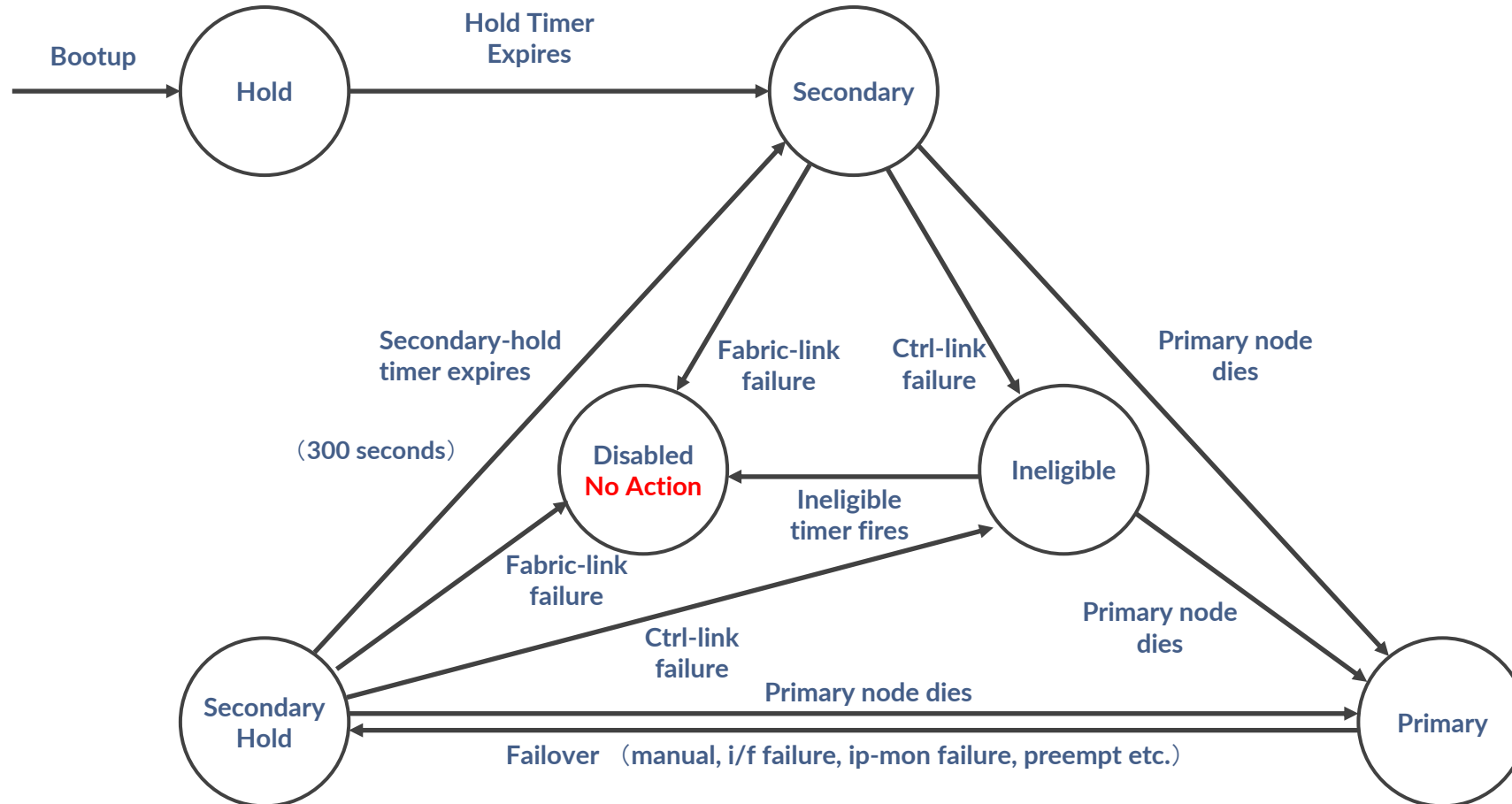
コントロールリンクモニタリング

- コントロールリンクモニタリング
 - コントロールリンクは、特に設定を加えることなく常にモニターされる
 - 然しながら、コントロールリンク リカバリー機能は、デフォルトでは設定されない
 - この設定は、セカンダリーノードが復旧した際に、自動でコントロールリンクを復旧させる機能
 - **30** 回のハートビート（デフォルトでは、**60** 秒）により正常性が確認できた後、セカンダリーノードをリブートさせる
 - コントロールリンクがダウンした時、セカンダリーノードは、**disable** のステータスになり、両方のノードが分離し別々に機能するのを防ぐ
 - コマンド： **set chassis cluster control-link-recovery**
- コントロールリンクがダウンした時、コントロールリンクを復旧させるには、コントロールリンク リカバリーの機能を利用するか、手動でセカンダリーノードをリブートするかのいずれかの方法が選択可能

ファブリックリンクモニタリング

- ファブリックリンクモニタリング
 - ファブリックリンクは、特に設定を加えることなく常にモニターされています。**Junos 10.4r4** 以降では、ファブリックリンクダウン発生から復旧時、リブートすることなく、モニタリングは再開される
 - ファブリックリンクは、最大 **2** 本まで冗長化することができます。**2** 本有効時、**1** 本は、**RTO** で利用し、残りの **1** 本は、実データを流すリンクとして利用する

SRX HA ステータス遷移



- **Disable** ステータスになるのは、セカンダリーノードのみ
- **Disable** ステートを復旧させるには、セカンダリーノードのリブートが必要
- 赤文字の「**No Action**」は、JUNOS 10.4r4 以降での動作

シャーシクラスタの無効化

- シャーシクラスタを無効化する場合
 - EPROM に書き込まれている内容をリセットする必要がある
 - 以下どちらかの手順で無効化（どちらも同じ効果）
 - Chassis Cluster を「 disable 」にして reboot

```
user@srx> set chassis cluster disable reboot
```

- または、Cluster ID を「 0 」に設定して reboot

```
user@srx> set chassis cluster-id 0 node 0 reboot
```



Appendix C: IPsec VPN の設定

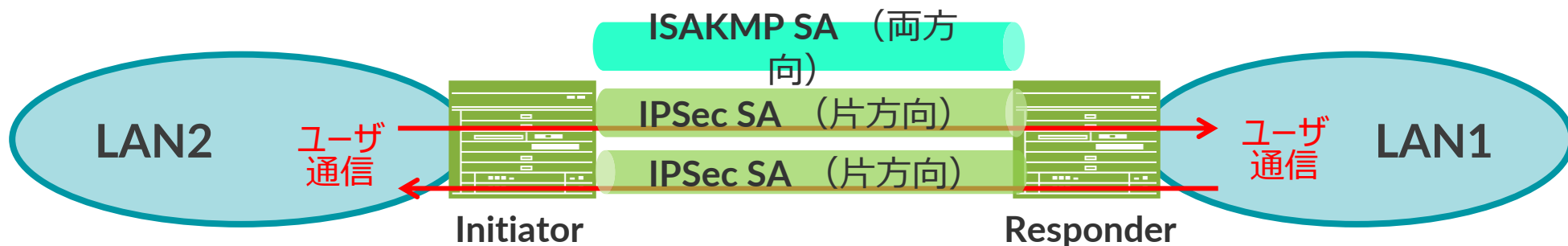
IPSec VPN とは

- **IPSec**

- 暗号技術を用いて IP パケット単位で改竄防止や秘匿機能を提供するプロトコル
- セキュリティゲートウェイ間で SA (Security Association) を作成
- ユーザトラフィックは SA 内を通過

- **IKE**

- 暗号/認証アルゴリズムの決定、暗号鍵交換のために利用されるプロトコル
- 2つのフェーズで SA を確立
 - IKE フェーズ 1: ISAKMP SA (双方向) を生成
 - IKE フェーズ 2: ユーザ通信が通過するための IPSec SA (片方向 x2) を生成
- IKE 折衝の開始側を Initiator、応答側を Responder と呼ぶ

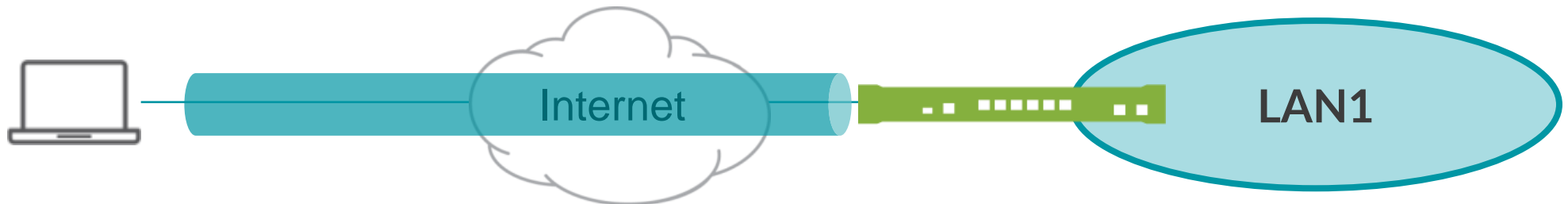


VPN 接続形態

- VPN 接続には大きく分けて下記の 2 通り
 - LAN 間接続
 - 離れた拠点間の LAN セグメント同士を VPN 接続



- リモートユーザ接続
 - セキュリティゲートウェイとユーザ端末間で VPN 接続
 - 端末側に VPN クライアントとなるソフトウェアが必要



LAN 間接続 IPSec VPN の設定方法

- SRX の LAN 間接続 VPN は、以下の 2 つの設定方法がある
 - ルートベース VPN
 - ルーティングにマッチする全トラフィックをトンネリング
 - ポリシーベース VPN
 - ポリシーにマッチするトラフィックのみをトンネリング

LAN 間接続 IPsec VPN 設定の手順

- **LAN 間接続 IPsec VPN** の設定は以下のステップで実施
 1. フェーズ **2** パラメータの設定
 - a. プロポーザルの設定
 - b. ポリシーの設定
 - c. ゲートウェイの設定
 2. フェーズ **1** パラメータの設定
 - a. プロポーザルの設定
 - b. ポリシーの設定
 - c. **VPN** の設定
- ルートベース **VPN** の場合
 - トンネルインタフェースの作成とゾーンの割り当て
 - ルーティングの設定
 - **VPN** へのバインディング
- ポリシーベース **VPN** の場合
 - トンネリングポリシーの作成

1-a. フェーズ 1 プロポーザルの設定

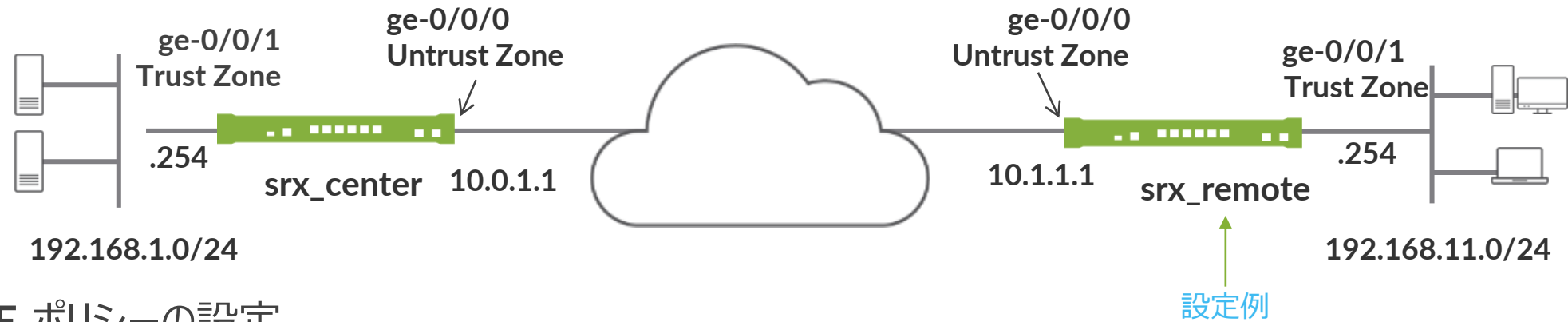
- ISAKMP SA のセキュリティ属性（プロポーザル）を定義
- 認証方式、鍵交換方式（DH group）、暗号化アルゴリズム、認証アルゴリズム等を指定

```
set security ike proposal IKE_PROPOSAL1 authentication-method pre-shared-keys
set security ike proposal IKE_PROPOSAL1 dh-group group2
set security ike proposal IKE_PROPOSAL1 authentication-algorithm sha1
set security ike proposal IKE_PROPOSAL1 encryption-algorithm aes-128-cbc
```

- パラメータの組合せが予め定義されており、こちらを利用することも可能

セット名	定義内容	表記
Basic	Proposal 1: Preshared key, DH g1, DES, SHA1 Proposal 2: Preshared key, DH g1, DES, MD5	pre-g1-des-sha pre-g1-des-md5
Compatible	Proposal 1: Preshared key, DH g2, 3DES, SHA1 Proposal 2: Preshared key, DH g2, 3DES, MD5 Proposal 3: Preshared key, DH g2, DES, SHA1 Proposal 4: Preshared key, DH g2, DES, MD5	pre-g2-3des-sha pre-g2-3des-md5 pre-g2-des-sha pre-g2-des-md5
Standard	Proposal 1: Preshared key, DH g2, 3DES, SHA1 Proposal 2: Preshared key, DH g2, AES128, SHA1	pre-g2-3des-sha pre-g2-aes128-sha

1-b、1-c. フェーズ 1 ポリシー、ゲートウェイの設定



- IKE ポリシーの設定
- 設定したプロポーザルを適用

```
set security ike policy IKE_POLICY1 proposals IKE_PROPOSAL1
set security ike policy IKE_POLICY1 pre-shared-key ascii-text juniper
```

- IKE ゲートウェイの設定
- IKE ポリシー、対向のアドレスとインタフェースを指定

```
set security ike gateway GW1 ike-policy IKE_POLICY1
set security ike gateway GW1 address 10.0.1.1
set security ike gateway GW1 external-interface ge-0/0/0
```

2-a. フェーズ 2 プロポーザルの設定

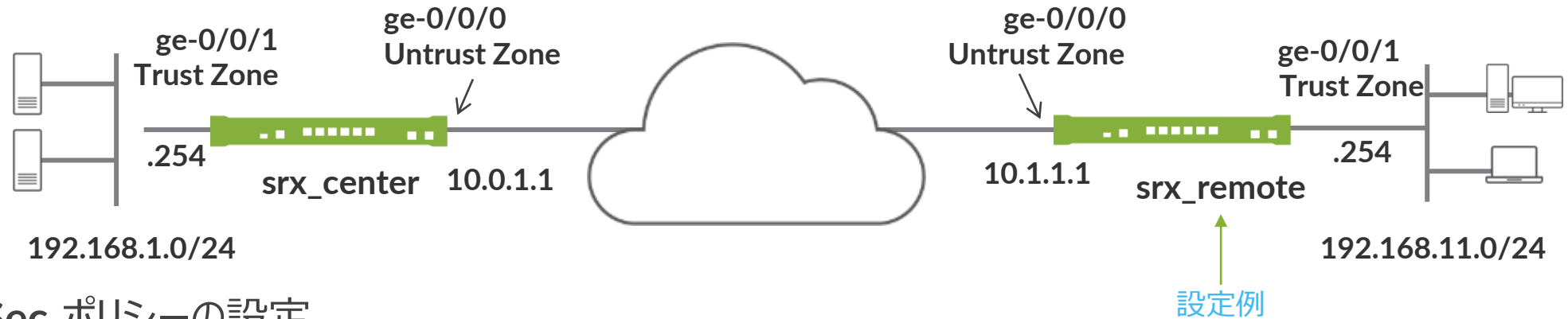
- IPsec SA のセキュリティ属性（プロポーザル）を定義
- プロトコル、暗号化アルゴリズム、認証アルゴリズム等を設定

```
set security ipsec proposal IPSEC_PROPOSAL1 protocol esp
set security ipsec proposal IPSEC_PROPOSAL1 authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROPOSAL1 encryption-algorithm aes-128-cbc
```

- パラメータの組合せが予め定義されており、こちらを利用することも可能

セット名	定義内容	表記
Basic	Proposal 1: no PFS, ESP, DES, SHA1 Proposal 2: no PFS, ESP, DES, MD5	nopfs-esp-des-sha nopfs-esp-des-md5
Compatible	Proposal 1: no PFS, ESP, 3DES, SHA1 Proposal 2: no PFS, ESP, 3DES, MD5 Proposal 3: no PFS, ESP, DES, SHA1 Proposal 4: no PFS, ESP, DES, MD5	nopfs-esp-3des-sha nopfs-esp-3des-md5 nopfs-esp-des-sha nopfs-esp-des-md5
Standard	Proposal 1: DH g2, ESP, 3DES, SHA1 Proposal 2: DH g2, ESP, AES128, SHA1	g2-esp-3des-sha g2-esp-aes128-sha

2-b、2-c. フェーズ 2 ポリシーの設定、VPN の設定



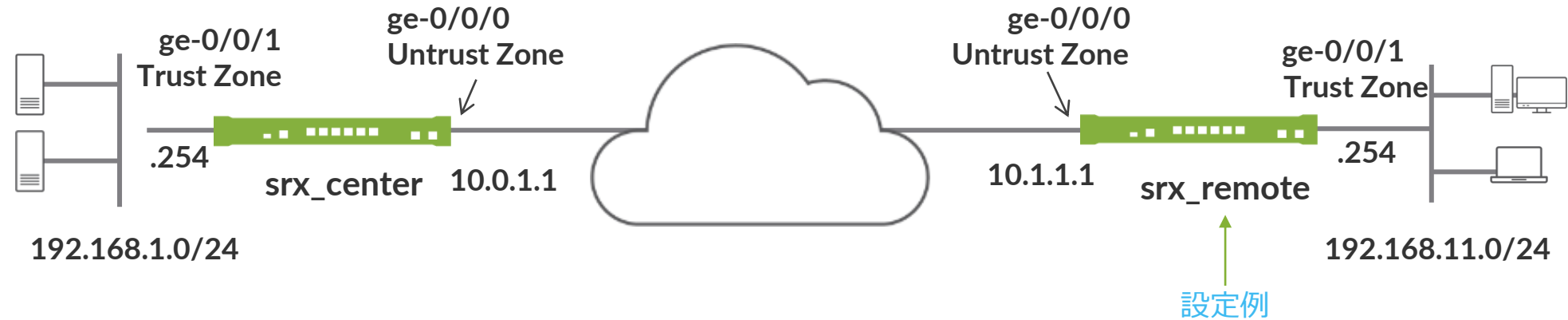
- IPsec ポリシーの設定
- 設定したプロポーザルを適用

```
set security ipsec policy IPSEC_POLICY1 proposals IPSEC_PROPOSAL1
```

- IPsec VPN の設定
- 設定済みのゲートウェイ、IPsec ポリシーを適用

```
set security ipsec vpn VPN1 ike gateway GW1  
set security ipsec vpn VPN1 ike ipsec-policy IPSEC_POLICY1  
set security ipsec vpn VPN1 establish-tunnels immediately
```

3. ルートベース VPN の設定



- トンネルインタフェースの作成

```
set interfaces st0 unit 0 family inet
```

- ルーティングの設定

```
set routing-options static route 192.168.1.0/24 next-hop st0.0
```

- IPsec VPN とのひもづけ

```
set security ipsec vpn VPN1 bind-interface st0.0
```

- Security Zone にアサイン

```
set security zones security-zone untrust interfaces st0.0
```


4. ポリシーベース VPN の設定

- アドレスブックの作成

```
set security zones security-zone trust address-book address Local-LAN 192.168.11.0/24
set security zones security-zone untrust address-book address Remote-LAN 192.168.1.0/24
```

- アクションが “Tunnel” のセキュリティポリシーを作成
- trust -> untrust

```
set security policies from-zone trust to-zone untrust policy 100 match source-address Local-LAN
set security policies from-zone trust to-zone untrust policy 100 match destination-address
Remote-LAN
set security policies from-zone trust to-zone untrust policy 100 match application any
set security policies from-zone trust to-zone untrust policy 100 then permit tunnel ipsec-vpn
VPN1
```

- untrust -> trust

```
set security policies from-zone untrust to-zone trust policy 200 match source-address Remote-LAN
set security policies from-zone untrust to-zone trust policy 200 match destination-address Local-
LAN
set security policies from-zone untrust to-zone trust policy 200 match application any
set security policies from-zone untrust to-zone trust policy 200 then permit tunnel ipsec-vpn
VPN1
```

※注意：ポリシーベース VPN とルートベース VPN の混在構成（設定）は不可

接続確認 – ISAKMP SA (フェーズ 1) の確認

```
user@SRX> show security ike security-associations
```

Index	State	Initiator cookie	Responder cookie	Mode	Remote Address
6706971	UP	845863c590392820	8ebfcc763b60a0de	Main	10.0.1.1

```
user@SRX> show security ike security-associations detail
```

```
IKE peer 10.0.1.1, Index 6706971, Gateway Name: GW1
```

```
Role: Responder, State: UP
```

```
Initiator cookie: 845863c590392820, Responder cookie: 8ebfcc763b60a0de
```

```
Exchange type: Main, Authentication method: Pre-shared-keys
```

```
Local: 10.1.1.1:500, Remote: 10.0.1.1:500
```

```
Lifetime: Expires in 25619 seconds
```

```
Peer ike-id: 10.0.1.1
```

```
Xauth user-name: not available
```

```
Xauth assigned IP: 0.0.0.0
```

```
Algorithms:
```

```
Authentication : hmac-sha1-96
```

```
Encryption : aes128-cbc
```

```
Pseudo random function: hmac-sha1
```

```
Diffie-Hellman group : DH-group-2
```

```
Traffic statistics:
```

```
Input bytes : 1148
```

```
Output bytes : 808
```

```
Input packets: 8
```

```
Output packets: 5
```

```
~~~~~
```

State : UP にならないと接続できていない
設定が対向側と同じになっているかを再チェック

接続確認 – IPsec SA (フェーズ 2) の確認

```
user@SRX> show security ipsec security-associations
Total active tunnels: 1
ID      Algorithm      SPI          Life:sec/kb  Mon lsys Port  Gateway
<131073 ESP:aes-cbc-128/sha1 7e4cac0d 2091/ unlim - root 500 10.0.1.1
>131073 ESP:aes-cbc-128/sha1 edfd7a93 2091/ unlim - root 500 10.0.1.1
```

IPsec SA は片方向なので
Inbound / Outbound の両方が
作成される

```
user@SRX> show security ipsec security-associations detail
```

```
ID: 131073 Virtual-system: root, VPN Name: VPN1
Local Gateway: 10.1.1.1, Remote Gateway: 10.0.1.1
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Version: IKEv1
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
Tunnel events:
  Sat Jul 16 2016 07:22:19: IPsec SA negotiation successfully completed (2 times)
  Sat Jul 16 2016 06:32:41: IKE SA negotiation successfully completed (1 times)
  Sat Jul 16 2016
  : Negotiation failed with error code NO_PROPOSAL_CHOSEN received from peer (2 times)
  Sat Jul 16 2016
  : Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
~~~~~
```

接続確認 - 暗号/復号トラフィックの統計確認

- IPsec SA 上での暗号化/復号化したバイト数、パケット数を表示

```
root@vSRX1> show security ipsec statistics
ESP Statistics:
  Encrypted bytes:          75696
  Decrypted bytes:         5208
  Encrypted packets:       498
  Decrypted packets:       62
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:           0
  Output packets:          0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

IPSec VPN トラブルシューティング

- IKE の Debug ログは、/var/log/kmd 内に保存
 - Debug 用設定

```
set security ike traceoptions flag all
set security ike traceoptions flag ike
```

- Debug ログ (kmd ファイル) の参照方法

```
user@SRX> show log kmd
```

- IKE Debug ログをリアルタイムにモニターする場合

```
user@SRX> monitor start kmd
user@SRX> monitor stop
```

- <http://kb.juniper.net/KB10100>
 - How to troubleshoot a VPN tunnel that is down or not active

IPSec 使用時の考慮点

- トンネルインタフェース（st0）の MTU 値はデフォルトで **9192**
ScreenOS と Route-based VPN を使用して接続する場合に問題となる場合がある
るので注意が必要



Appendix D: NAT Pool Options

Source NAT with address-persistent

- NAT 動作時に特定のホストにセッションにかかわらず同じ IP Address を Pool から割り当てる
 - 特定のホストから最初にセッションがイニシエートされ NAT のアドレスが割り当てられると、以降そのホストから複数のセッションがイニシエートされても、同一の IP Address を NAT で割り当てることが可能

```
set security nat source address-persistent
```

- Global 設定となるため、設定を投入するとすべての Pool に反映される

アドレスプール設定補足

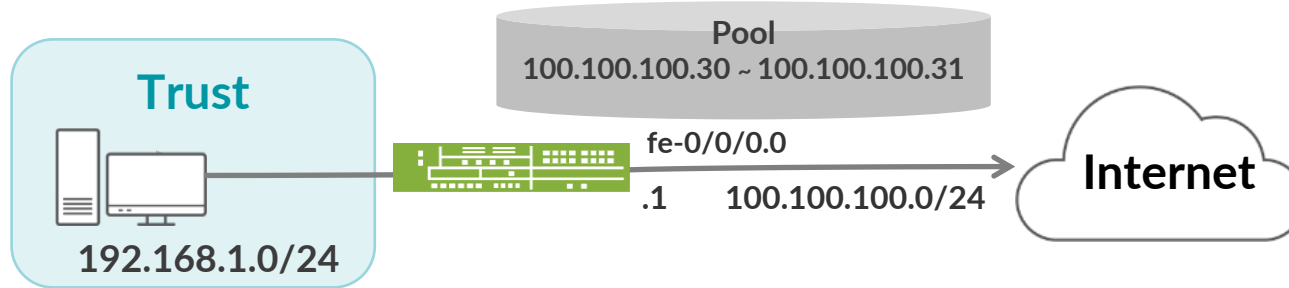
- アドレスプールの形態
 - 単一の IP アドレス
 - IP アドレスレンジ
 - インタフェース (**Source NAT** のみ)
- オプション
 - ポート変換オフ (**port no-translation**)
 - **Overflow Pools**
 - プールのアドレスを使い切った場合のフォールバック用に設定
 - インタフェースアドレスを使用
 - 別のプールを参照
 - ポート変換なしの **pool** が必要
 - アドレスシフト

Source NAT Pool の設定例

```
lab@srx# show security nat
source {
    pool src_nat_pool_napt {
        address {
            100.100.100.20/32 to 100.100.100.29/32;
        }
        port {
            no-translation;
        }
        overflow-pool interface;
    }
}
```

アドレスプール動作の確認

- PAT 動作の無効化 (port no-translation)
- プール超過時 NAPT (overflow-pool interface)



```
lab@srx> show security flow session
Session ID: 11120, Policy name: trust-to-untrust/4, Timeout: 1580, Valid
  In: 192.168.1.22/21003 --> 100.100.100.254/23;tcp, If: vlan.0, Pkts: 36, Bytes: 1481
  Out: 100.100.100.254/23 --> 100.100.100.30/21003;tcp, If: fe-0/0/0.0, Pkts: 36, Bytes: 1523

Session ID: 11127, Policy name: trust-to-untrust/4, Timeout: 1790, Valid
  In: 192.168.1.23/1267 --> 100.100.100.254/22;tcp, If: vlan.0, Pkts: 17, Bytes: 1673
  Out: 100.100.100.254/22 --> 100.100.100.31/1267;tcp, If: fe-0/0/0.0, Pkts: 18, Bytes: 1767

Session ID: 11159, Policy name: trust-to-untrust/4, Timeout: 1794, Valid
  In: 192.168.1.24/1044 --> 100.100.100.254/80;tcp, If: vlan.0, Pkts: 43, Bytes: 40039
  Out: 100.100.100.254/80 --> 100.100.100.1/64506;tcp, If: fe-0/0/0.0, Pkts: 43, Bytes: 40039
```

NAT されているが
ポート変換されていない

Pool を超過したため
IP アドレスで NAPT されている

Source NAT with address-shifting

- NAT 動作時に Private : Public が 1 : 1 でマッピングされる
 - Host-address-base で基点になる Private アドレスを設定

```
set security nat source pool A address 192.168.1.1/32 to 192.168.1.20/32
set security nat source pool A host-address-base 10.1.1.5/24
```

- show security nat source pool all コマンドで確認
 - 10.1.1.5 ~ 25 が 192.168.1.1 ~ 20 と 1 : 1 に対応

```
root> show security nat source pool all
node0:
-----
Total pools: 1
Pool name      : A
Pool id       : 4
Routing instance : default
Host address base : 10.1.1.5
Port          : no translation
Port overloading : 0
Address assignment : static-paired
Total addresses : 20
Translation hits : 0
Address range   Single Ports Twin Ports
192.168.1.1 - 192.168.1.20      0           0
```

Source NAT with port-overloading-factor

- **port-overloading-factor** を “N” と設定することで、内部的に **PAT** のために使用するポート番号を **64k×N** の数まで増やして使用することが可能

```
set security nat source pool src_nat_pool_napt address 100.100.100.30/32
set security nat source pool src_nat_pool_napt port port-overloading-factor 2
```

- **N=1** というのがデフォルトの状態
- **port-overloading-factor** の設定としてできる値は **1** から **32** まで
- **Port-overloading-factor** を利用できる **Pool Address** の数はプラットフォーム毎に制限が定められている
 - **Branch SRX Series = 1**
 - **SRX5k Series = 128** (15.1X49-D40 ~、それ以前は **16**)



Appendix E: Security Logging

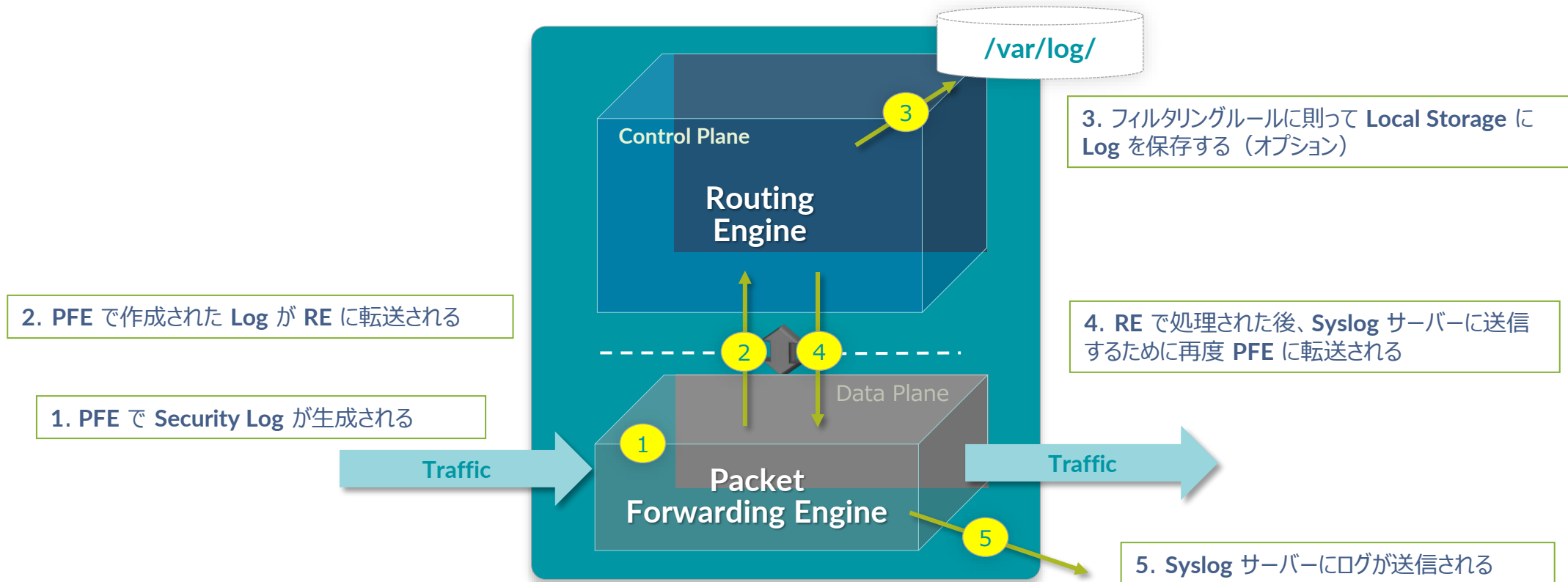
Security Logging

- Junos のシステムにて利用される通常のシステムログとは別に、トラフィックログ（**Security Logging**）を取得することが可能
 - **Security Logging** は 2 つのフォーマットから選択が可能
 - 通常の Syslog（RFC3164）
 - **Structured Syslog**
 - より詳細なセキュリティ情報を取得したい場合に使用
 - **Security Logging** は 2 つの収集方法から選択が可能
 - **Event Mode**
 - **Default** 設定（最大 1500 event/sec ※）※ただしロギングパフォーマンスはプラットフォームに依存
 - **Stream Mode**
 - 高負荷なトラフィック環境で **Security Log** の取得が必要な場合には推奨されるモード

Security Logging

• Event Mode

- Security Log は一度 Routing Engine で処理した後に Syslog サーバへ送信されるため、高トラフィック時には Routing Engine の処理負荷が増大するのでデザインに検討が必要
- 一方で、Security Log のフィルタリングや内部 Storage への保存が可能な方式



Security Logging

• Event Mode

```
set security log mode event
set security log event-rate 100
set security log format sd-syslog
```

Event Mode を宣言して、イベントレート、フォーマットなどを指定

```
set system syslog host 192.168.0.99 any any
set system syslog host 192.168.0.99 match RT_FLOW
```

Traffic Log のメッセージは "RT_FLOW" にマッチする

```
set system syslog file TRAFFIC-LOG any any
set system syslog file TRAFFIC-LOG match RT_FLOW
```

Syslog サーバーに送信する場合はHostを指定

```
set security policies from-zone trust to-zone untrust policy trust-to-untrust match source-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match destination-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match application any
set security policies from-zone trust to-zone untrust policy trust-to-untrust then permit
set security policies from-zone trust to-zone untrust policy P1 match source-address any
set security policies from-zone trust to-zone untrust policy P1 match destination-address any
set security policies from-zone trust to-zone untrust policy P1 match application any
set security policies from-zone trust to-zone untrust policy P1 then permit
set security policies from-zone trust to-zone untrust policy P1 then log session-init
```

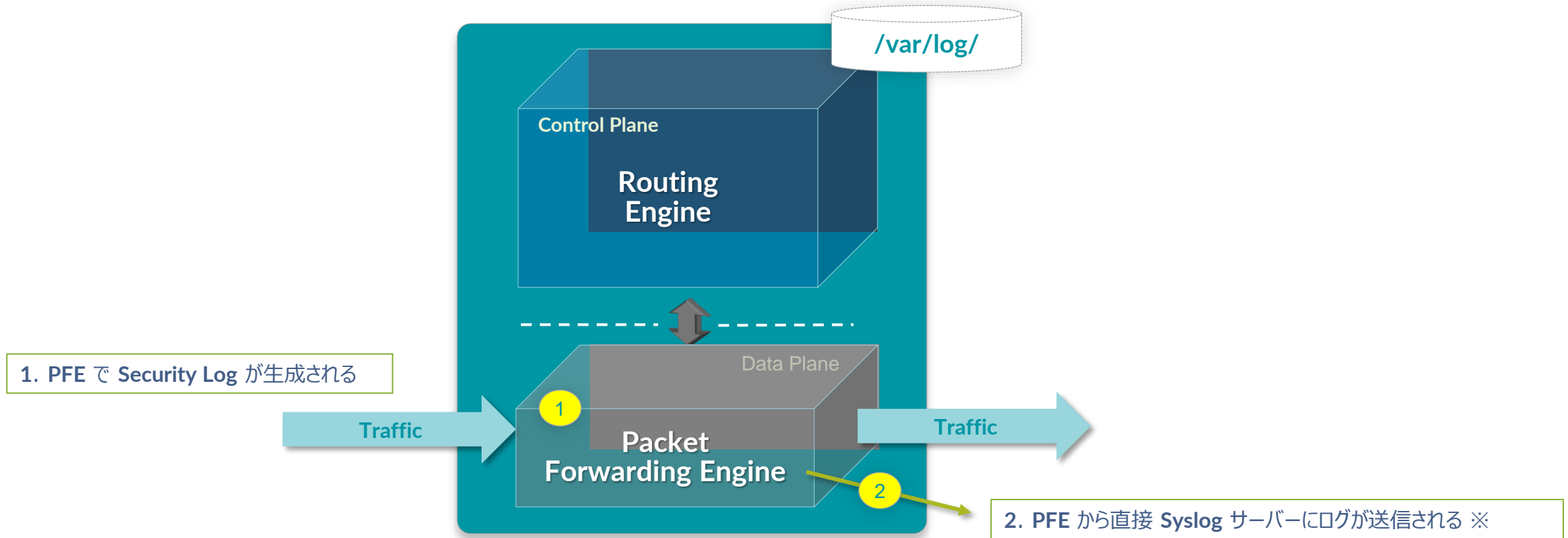
Log を Local Storage に保存する場合は File 名を指定

Security Log を取得したい FW ポリシーでアクションを指定

Security Logging

- Stream Mode

- Security Log は、Packet Forwarding Engine 内で処理され、Syslog サーバーへ転送される
(高い Logging Rate を期待することができますが、Local Storage への Log 保存などは行えない)



※Stream Mode 使用時には、Syslog Server への Log 送信は Revenue Port から送信される必要がある (FXPO からの送信は未サポート)

Security Logging

- Stream Mode

```
set security log mode stream
set security log source-address 192.168.0.254
set security log stream TRAFFIC-LOG format sd-syslog
set security log stream TRAFFIC-LOG host 192.168.0.99
```

Stream Mode を宣言して、Source Address、フォーマット、Syslog サーバーのターゲットなどを指定

```
set security policies from-zone trust to-zone untrust policy trust-to-untrust match source-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match destination-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match application any
set security policies from-zone trust to-zone untrust policy trust-to-untrust then permit
set security policies from-zone trust to-zone untrust policy P1 match source-address any
set security policies from-zone trust to-zone untrust policy P1 match destination-address any
set security policies from-zone trust to-zone untrust policy P1 match application any
set security policies from-zone trust to-zone untrust policy P1 then permit
set security policies from-zone trust to-zone untrust policy P1 then log session-init
```

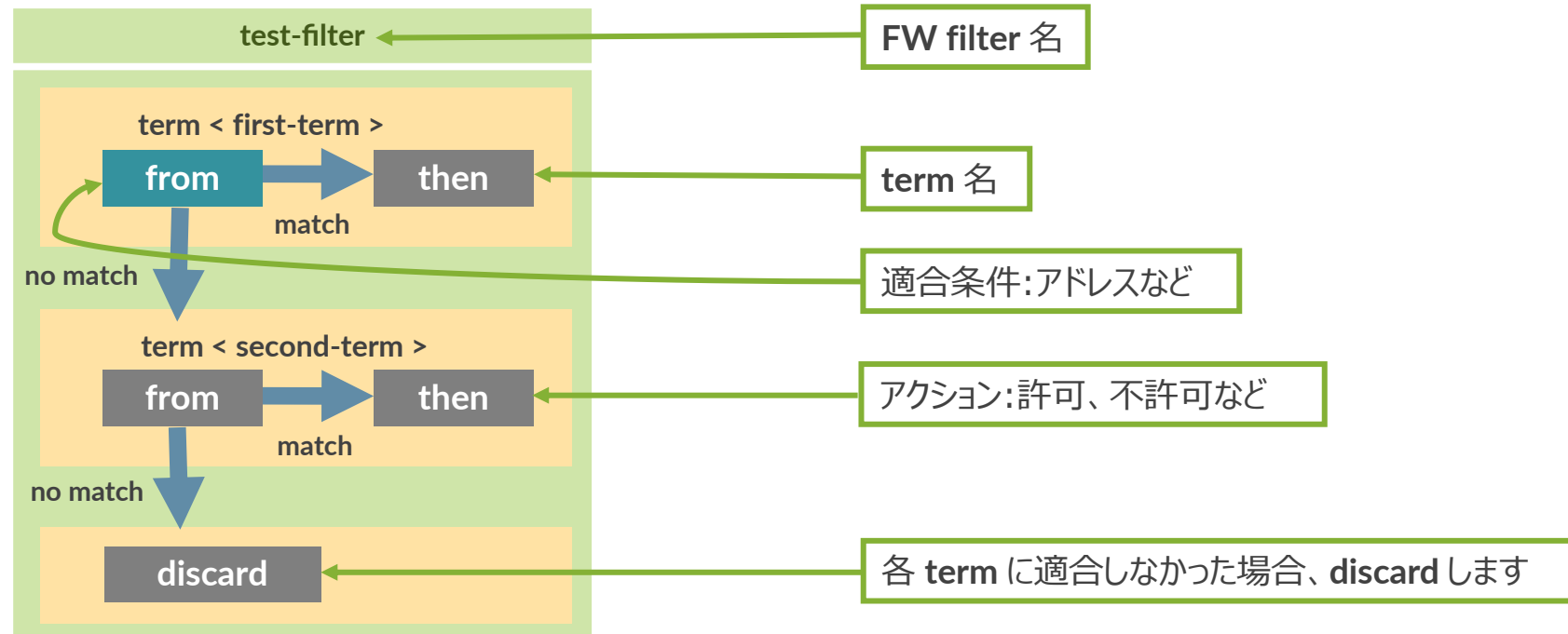
Security Log を取得したい FW ポリシーでアクションを指定



Appendix F : 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
term BLOCK {
  from {
    source-address {
      10.10.10.0/24;
    }
  }
  then {
    discard;
  }
}
term PERMIT {
  then accept;
}
```

FW filter 名

term 名

適合条件: 10.10.10.0/24 からの通信

アクション: 不許可

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

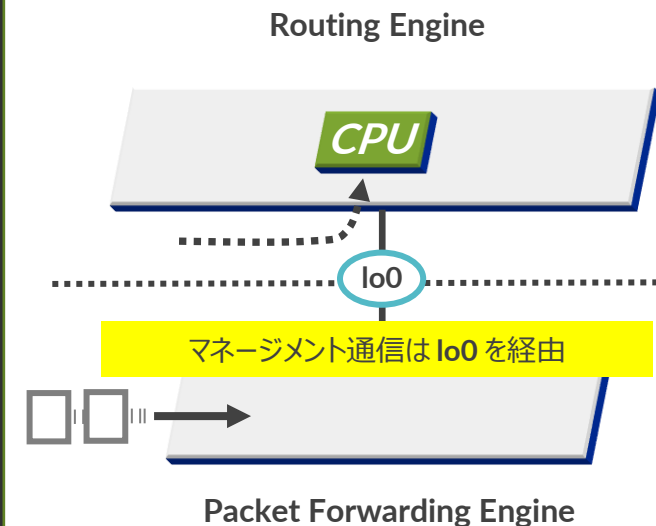
1. FW フィルタを作成

- 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)



THANK YOU

JUNIPER
NETWORKS | Driven by
Experience™