

# Junos® 基本シリーズ

## Day One: IOSエンジニア向けJunos解説

著者: クリス・ジョーンズ

第1章: 基本事項.....	5
第2章: 基本設定.....	17
第3章: ケーススタディ.....	65

© 2012 by Juniper Networks, Inc. All rights reserved.

Juniper Networks、Juniper Networks の ロゴ、Junos、NetScreen、および ScreenOS は、Juniper Networks, Inc. (以下、ジュニパーネットワークス) の米国およびその他の国における登録商標です。Junose は、ジュニパーネットワークスの商標です。その他の商標、サービスマーク、登録商標、登録サービスマークは、それぞれの所有者に帰属します。

ジュニパーネットワークスは、本書に誤りが含まれることがあっても責任を負いません。ジュニパーネットワークスは、予告なく本書を変更、修正、転載、または改訂する権利を留保します。ジュニパーネットワークスが製造、販売する製品、あるいはその部品は、ジュニパーネットワークスが保有する、あるいはライセンスを受けた以下の米国特許のうち 1 件または複数により保護されている場合があります。米国特許第 5,473,599 号、第 5,905,725 号、第 5,909,440 号、第 6,192,051 号、第 6,333,650 号、第 6,359,479 号、第 6,406,312 号、第 6,429,706 号、第 6,459,579 号、第 6,493,347 号、第 6,538,518 号、第 6,538,899 号、第 6,552,918 号、第 6,567,902 号、第 6,578,186 号、第 6,590,785 号。

**発行者：Juniper Networks Books**

著者：クリス・ジョーンズ

貢献者：ダナ・ヤンク

技術校閲者：トム・ドワイヤー、ニック・ライス、ステファン・フォアント

主編集者：パトリック・エイムズ

原稿整理・校正編集者：ナンシー・ケルベル

J-Net コミュニティ管理者：ジュリー・ワイダー

#### 著者の紹介

クリス・ジョーンズは、ジュニパーネットワークスの認定 JNCIE-ENT #272 と Cisco Systems の認定 CCIE #25655 (R&S) を受けたネットワークエンジニアです。業界での経験は 8 年を越え、Cisco とジュニパーネットワークスの両方の製品・ソリューションに携わっています。

#### 著者の謝辞

長期間に及ぶ本書の執筆期間、あらゆる面でサポートしてくれた私の妻、ジェニファーに感謝しています。CCIE と JNCIE の取得に向けて学習に専念しているときに、忍耐強く私を支えてくれたことにも、今さらですが感謝しています。本書のために IOS の設定を提供してくれた私の親友であるダナ・ヤンク氏には、特に感謝しています。彼の支援がなければ、本書は完成しなかったでしょう。最後に、本書を執筆する機会を与えてくれた編集者のパトリック・エイムズ氏にも感謝しています。本書が無事に完成したのは、彼の適切なアドバイスのおかげです。

ISBN : 978-1-936779-54-3 (書籍)

印刷 : Vervante Corporation (米国)

ISBN : 978-1-936779-55-0 (電子書籍)

改訂 : 初版、2012 年 8 月

2 3 4 5 6 7 8 9 10 #7100157-en

本書はさまざまな形式で [www.juniper.net/dayone](http://www.juniper.net/dayone) から入手できます。

## 貢献者と技術校閲者の紹介

本書のような書籍が客観的に記述されていることはまれなケースであり、本当に役に立つと言われても、そうではないことが実際のところです。ただし、本書については、以下の貢献者の方々の協力により、可能な限り、内容は簡単かつ客観的に記述されています。

- **ダナ・ヤンク氏は**、Insight Enterprises社のネットワーク技術者であり、CCIE #25567およびJNCIP-ENTを取得しています。本書の執筆に際しての第一の貢献者は彼であり、独自の情報源として、Cisco IOSの設定全般を提供してくれました。
- **トム・ドワイヤー氏は**、Nexum Inc.社のシニアネットワークセキュリティエンジニアです。ネットワーク分野で18年の経験を積み、JNCIP-ENT/SECやJNCIなど、ジュニパーネットワークスの複数の認定を保持しています。
- **ニック・ライス氏は**、英国Pulsant社のシニアネットワークエンジニアです。ジュニパーネットワークスの認定JNCIE-ENT #232を取得し、IT業界での経験は7年を越えます。技術校閲者として、本書の執筆に協力してくれました。
- **ステファン・フォアント氏は**、ジュニパーネットワークスのテクニカルトレーナー兼JNCP試験委員です。JNCIEのトリプル認定を受けた数少ない1人であり、両OSの技術校閲を担当してくれた彼の協力のおかげで、本書は完成しました。

## IOS と Junos の比較

編集者、貢献者、著者の中で話し合いを重ねた結果、2つのOSを単純に比較する形で紹介するということになりました。具体的には、少数の項目ごとに、交互に例を提示して比較しています。本書の読者は、Ciscoの設定（すでに知識があるという前提）に目を通してから、対応するJunosの設定を注釈と相違点の説明とともに確認してください。

本書の最初の章では両OSに共通するタスクについて説明し、次の章では両OSで一般的な設定を比較し、さらに最後の章ではシンプルなネットワーク全体の設定を比較して結論をまとめています。

本書は、IOSエンジニアがジュニパーネットワークス製デバイスを管理する上で必要な知識を1日で身につけるためのガイドブックです。

## 本書を読む前に知っておくべきこと

- 本書の目的を達成するには、Junos OSの基本知識が必要です。本書はIOSとJunosで一般的な設定を比較することを目的としていますが、Junos CLI操作自体の説明は取り上げていません。
- 本書は、読者がJunosに関する2冊のDay Oneシリーズの書籍、『*Day One: Junos CLIの探究*』と『*Day One: Junosの基本設定*』を読み終えて、内容を理解していることを前提としています。この両書は、<http://www.juniper.com/dayone>から無償でダウンロードできます。
- 本書は、読者がIOSを熟知し、基本的な管理タスクの設定に加えて、シンプルなIGPやBGPのシナリオにも精通していることを前提としています。
- さらに、OSPFやBGPなど、一般的なネットワークプロトコルを理解していることも前提としています。

## 本書の学習目標

- JunosとIOSで思考プロセスが異なることを理解する
- Junosルーターでシンプルな機能と一般的なタスクを設定する
- 一般的なJunosの設定とIOSの設定を比較する
- VLAN、OSPF、およびBGPを使用して小規模なネットワーク用の設定を作成する

## Day One へようこそ

Day One シリーズの書籍は、新しいテーマにすぐに取りかかれるように、作業の初日に必要となる情報だけをまとめたものです。Day One シリーズでは、ステップごとに手順を説明し、そのまま使えるような詳しい例を数多く掲載しています。さらに詳しい解説が記載された資料についても提示しています。Day One ライブラリの詳細については、<http://www.juniper.net/dayone> を参照してください。

# 第1章

## 基本事項

ホスト名の設定.....	7
ユーザーの作成.....	8
SSHv2の有効化.....	10
インタフェースへのIPアドレスの割り当て.....	11
設定の保存.....	13
ソフトウェアのアップグレード.....	15
結論.....	16

本書は、IOS エンジニアが一般的なネットワークタスクを Junos の同等の機能と比較して考える上で参考になるガイドです。最初はタスクの内容が初歩的すぎると思われるかもしれませんが、新しい言語を学習する場合と同様に、まずは一般的な日常の事柄から始めていきましょう。

さらに詳しくは 前付け部分の説明の繰り返しになりますが、本書は、読者が Junos の基本事項を理解していることを前提としています。Day One ライブラリの『*Junos CLI の探究*』と『*Junos の基本設定*』の両書の内容を理解しておいてください。この両書は <http://www.juniper.com/dayone> から無償でダウンロードできます。また、iTunes の iBookstore と Amazon の Kindle ブックストアでも入手できます。

この章では、エンジニアがジュニパーネットワークス製ルーターを設定する場合に必要な、以下のような基本的なタスクを取り上げています。

- ホスト名の設定
- ユーザーの作成
- SSHv2の有効化
- インタフェースへのIPアドレスの割り当て
- 設定の保存
- ソフトウェアのアップグレード

## ホスト名の設定

ホスト名は、ルーターに設定できる最も一般的な設定の1つです。ただし、通常の設定時やログの取得時にデバイスを識別する場合には、不可欠な情報です。それぞれの OS の設定を比較しましょう。

### IOS の設定

#### IOS デバイスでホスト名を設定するには

```
Router#configure terminal  
Router(config)#hostname R1  
R1(config)# end  
R1#
```

#### IOS デバイスで設定済みのホスト名を確認するには

新しいホスト名はルーターのプロンプトに表示されますが、実行中の設定 (running-config) でも設定済みのホスト名を確認できます。

```
R1#show running-config | include hostname  
hostname R1  
R1#
```

### Junos の設定

#### Junos デバイスでホスト名を設定するには

```
cjones@router> configure  
cjones@router# set system host-name R1  
cjones@router# commit and-quit  
cjones@R1>
```

#### Junos デバイスで設定済みのホスト名を確認するには

繰り返しになりますが、新しいホスト名はルーターのプロンプトに表示されます。設定済みのホスト名は、設定内でも確認できます。

```
cjones@R1> show configuration system | match host-name  
host-name R1;
```

### まとめ

これまで見てきたように、ホスト名の設定はコマンドを1つ実行するだけであり、IOS と Junos でもほとんど同じです。ここまでは、特に問題はありません。

2つの OS のルックアンドフィールの違いについて、きちんと理解できていることでしょう。本書を読み終えたときには、この違いの理由を理解して、余裕を持って対処できるようになるはずです。この違いに注目しながら、次のタスクに進みましょう。

## ユーザーの作成

ユーザーの作成は、設定の重要な部分を占めています。複数のエンジニアに対して、設定の表示および修正に必要なアクセス権を付与する場合を比較しましょう。

### IOS の設定

*IOS で平文パスワードを使用してユーザーを作成するには*

```
R1#configure terminal
R1(config)#username plaintextuser password p@$sw0Rd
R1(config)#end
R1#
```

*IOS で暗号化パスワードを使用してユーザーを作成するには*

```
R1#configure terminal
R1(config)#username encrypteduser secret p@$sw0Rd
R1(config)#end
R1#
```

*IOS で作成済みのユーザーを確認するには*

```
R1#show running-config | include username
username plaintextuser password 0 p@$sw0Rd
username encrypteduser secret 5 $1$kbb9$i/TL0SPZr.F8g2LD/MFVi0
R1#
```

### Junos の設定

*Junos でユーザーを設定するには*

```
cjones@R1> configure
cjones@R1# set system login user bob class super-user full-name "Bob" authentication
plain-text-password
New password:
Retype new password:

cjones@router# commit and-quit
cjones@R1>
```

*Junos でユーザーの作成を確認するには*

```
cjones@R1> show configuration system login
user bob {
    full-name Bob;
    uid 2001;
    class super-user;
    authentication {
        encrypted-password "$1$wkvA6Ppe$NHN5HgacMAE40Y1vMxF7j/"; ## SECRET-DATA
    }
}
```



```
}
```

```
cjones@R1>
```

## まとめ

Junos のユーザー設定は IOS の場合よりも若干、複雑ですが、そのままでも柔軟性は大幅に上回っています。定義済みのユーザーロールを割り当てる機能は、明らかなメリットです。

Junos では、パスワードがデフォルトでハッシュ化されている点にも注意してください。

ここで少し時間を取って、Junos の設定表示について説明しましょう。この例は、中かっこ付きの標準的な Junos の設定出力を示しています。(C 言語のプログラミングとは異なるので) 最初はわかりにくいかもしれませんが、実際の形式をいったん理解してしまえば、設定の各セクションが明確な区別もなく並んでいる IOS の場合よりも、はるかに直観的であることがわかります。また、設定の規模が大きくなるほど、この出力形式のメリットは明らかになります。

**Junos のヒント** これに対して、場合によっては、入力そのままの状態ですべての set コマンドのリスト形式で設定を表示できると役に立つことがあります。この表示に切り替えるには、show configuration の後に | display set を続けて入力するだけです。

```
cjones@R1> show configuration | display set
set system login user bob full-name Bob
set system login user bob uid 2001
set system login user bob class super-user
set system login user bob authentication encrypted-password "$1$wkvA6Ppe$NHN5HgacMAE4
0Y1vMxF7j/"
```

この操作は、編集モードでも実行できます。具体的には、configuration キーワードを省略するだけです。

```
cjones@R1# show | display set
set system login user bob full-name Bob
set system login user bob uid 2001
set system login user bob class super-user
set system login user bob authentication encrypted-password "$1$wkvA6Ppe$NHN5HgacMAE4
0Y1vMxF7j/"
```

## SSHv2の有効化

SSHv2は、Telnetのような非暗号化通信ではなく、暗号化通信を使用する認証プロトコルです。2つのOSで、この基本的なタスクを比較しましょう。

### IOSの設定

#### *IOSでSSHv2を設定するには*

```
R1#configure terminal
R1(config)#ip domain-name juniper.net
R1(config)#crypto key generate rsa modulus 1024
R1(config)#ip ssh version 2
R1(config)#end
R1#
```

#### *IOSでSSHv2を確認するには*

```
R1#show ip ssh
SSH Enabled - version 2.0
Authentication timeout:120 secs; Authentication retries:3
R1#
```

### Junosの設定

#### *JunosでSSHv2を設定するには*

```
cjones@R1> configure
cjones@R1# set system services ssh
cjones@R1# commit and-quit
cjones@R1>
```

#### *JunosでSSHv2を確認するには*

```
cjones@R1> show configuration system services
ssh;
```

### まとめ

JunosがFreeBSDを基盤としているおかげで、OSのインストール時にはRSAキーがすでに生成されているので、JunosでSSHv2を設定する方が手順としては簡単です。これに対して、IOSでは、RSAキーを生成する前に、ドメイン名を設定しておく必要があります。

IOSの場合はSSHのデフォルトバージョンが1.99であるのに対して、Junosの場合はバージョン2です。

注 SSHトラフィックをTCP/22で許可するため、適用済みのファイアウォールフィルタがあれば、修正する必要があります。

## インタフェースへの IP アドレスの割り当て

デバイスが通信するには IP アドレスが必要であり、IP アドレスの割り当ては、ネットワーク設定の種類を問わず、基本的なタスクです。ただし、この基本的なタスクは、2つの OS の違いを見ていく上で、わかりやすい違いをいくつか示しています。

### IOS の設定

#### IOS のインタフェースで IP アドレスを設定するには

```
R1#configure terminal
R1(config)#interface f0/0
R1(config-if)#ip address 192.0.2.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#end
R1#
```

#### IOS のインタフェースで IP アドレスを確認するには

```
R1#show ip interface brief FastEthernet 0/0
Interface                IP-Address      OK?Method Status      Protocol
FastEthernet0/0         192.0.2.1       YES manual  up          up
```

### Junos の設定

#### Junos のインタフェースで IP アドレスを設定するには

この Junos の例の構文では、以下の複数の要素をすべて1つのコマンドで指定しています。

- インタフェース名 (ge-0/0/0)
- 論理ユニット番号 (この例では、0)
- アドレスファミリ (inet – IPv4アドレス用に使用されるファミリ)
- インタフェースアドレス自体 (192.0.2.1/24)

**注** Junos では、インタフェースの論理ユニット番号は、IOS のサブインタフェースと基本的に同義です。ただし、Junos では、一貫性を確保するためユニット番号が常に必要です。論理設定 (たとえば、IP アドレス) は、ユニットに帰属します。物理設定 (たとえば、速度や二重化) は、階層の物理インタフェースに帰属します。VLAN タギングなど、特別な設定がなければ、ユニット番号は通常、0 です。

```
cjones@R1> configure
cjones@R1# set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
cjones@R1# commit and-quit
cjones@R1>
```

### Junos のインタフェースで IP アドレスを確認するには

```
cjones@R1> show configuration interfaces fe-0/0/0
unit 0 {
    family inet {
        address 192.0.2.1/24;
    }
}
```

```
cjones@R1> show interfaces terse fe-0/0/0
Interface          Admin Link Proto  Local          Remote
fe-0/0/0           up    up
fe-0/0/0.0        up    up  inet    192.0.2.1/24
```

### まとめ

Junos のインタフェースで IP アドレスを設定する場合は、1つのコマンドだけで済みます。ドット区切りの 10 進数形式でサブネットマスクを入力する必要がある IOS の場合とは異なり、Junos OS では、CIDR 表記を使用できます。

**注** Junos では、サブネットマスクを指定しなければ、常に /32 アドレスが設定されます。

IOS のインタフェースで追加の IP アドレスを設定する場合、secondary キーワードを追加していなければ、設定済みのアドレスは上書きされます。これに対して、Junos のインタフェースで追加のアドレスを設定した場合は、インタフェースには元のアドレスと新しいアドレスの両方が設定されます。

IOS では管理上、デフォルトでインタフェースがシャットダウンされ、no shutdown コマンドでインタフェースを有効化する必要があります。Junos では、インタフェースはデフォルトで有効化されており、set interfaces <interface> disable とコミットによってシャットダウンすることが可能です。

## 設定の保存

設定の保存は、ルーターやスイッチを管理する上で重要な意味があります。デバイスを再起動した場合、それまでに保存していない設定は失われます。この基本的なタスクは、2つのOSでコミットの実行方法が異なることを示しています。

## IOS の設定

### IOS で設定を保存するには

```
R1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#
```

### IOS で保存済みの設定を確認するには

設定を確認するには、nvram:ファイルシステムにstartup-configファイルがゼロ以外のサイズで存在することをチェックする必要があります。

```
R1#dir nvram:
Directory of nvram:/

 52  -rw-          839          <no date>  startup-config
 53  ----         1924          <no date>  private-config
   1  ----          15          <no date>  persistent-data
   2  -rw-          0          <no date>  ifIndex-table

57336 bytes total (52473 bytes free)
R1#
```

## Junos の設定

Junosでは、アクティブな設定は、デバイスが起動したときにロードされる設定と同じです。

Junosで候補設定を保存するには、commitコマンドを使用します。コミットによって候補設定がアクティブ化され、変更が適用される前に設定に問題があれば通知されます。

commit checkコマンドを使用すると、実際にはコミットせずに、設定を検証することも可能です。

最後に、commit confirmedコマンドを使用すると、指定の時間が経過してから、ルーターを直前の正常な設定にロールバックすることが可能です。このコマンドは、ファイアウォールフィルタやルーティングポリシーに対して大幅な変更をコミットする場合に特に便利です。ソフトウェアの設定のロールバックを防ぐには、commitの確認時間（デフォ

ルトでは、10分) が経過するまでに、単純に `commit` をもう1回、入力します。

### Junos で設定を保存するには

```
cjones@R1> configure
cjones@R1# commit and-quit
cjones@R1>
```

### Junos で保存済みの設定を確認するには

保存済みの設定を確認する最適な方法は、前のバージョンと比較することです。この例では、ロールバック番号 0 (現在の設定) をロールバック番号 1 (以前にコミットした設定) と比較します。

```
cjones@R1> show system rollback compare 1
[edit interfaces]
- fe-0/0/0 {
-     unit 0 {
-         family inet {
-             address 192.0.2.1/24;
-         }
-     }
- }
```

### まとめ

IOS では、入力したコマンドは即座に有効になります。設定を保存するには、実行中の設定を起動設定にコピーする必要があります。これに対して、Junos では、*commit model* を使用すると、デバイスに即座に影響を及ぼさずに、設定コマンドを入力することが可能です。

Junos では、`confirmed` キーワードを指定して、設定を自動的にロールバックすることも可能です。IOS では、起動設定にロールバックするには、ルーターを設定する必要があります。



## アップグレード済みの IOS ソフトウェアを確認するには

```
R1#show version | include IOS
Cisco IOS Software, 2600 Software (C2600-ADVENTERPRISEK9-MZ), Version 12.4(12)T9,
RELEASE SOFTWARE (fc5)
R1#
```

## Junos の設定

### Junos ソフトウェアをアップグレードするには

Junos ルーターでのソフトウェアのアップグレードは、とてもシンプルです。FreeBSD を基盤としているおかげで、ソフトウェアパッケージをルーターに移動する作業はとてもシンプルです。最初のステップでは、SCP、FTP、または TFTP を使用して、ソフトウェアパッケージを /var/tmp ディレクトリにコピーします（別の場所にも配置できますが、一般的な場所は /var/tmp です）。次に、以下のコマンドをそのまま実行します。

```
cjones@R1> request system software add reboot /var/tmp/ junos-srxsme-10.4R7.5-
domestic.tgz
```

デバイスが再起動して、新しいソフトウェアがインストールされます。

### アップグレード済みの Junos ソフトウェアを確認するには

```
cjones@R1> show version
Hostname:R1
Model: srx100h
JUNOS Software Release [10.4R7.5]
```

## まとめ

IOS ソフトウェアと Junos ソフトウェアのどちらもアップグレードは簡単であり、コマンドを1つ実行するだけです。ただし、IOS の場合、別に reload コマンドも実行する必要があります。

## 結論

2つのOSのルックアンドフィールの違いについて、理解できたでしょうか。覚えておくべきポイントとして、Junosでは、show configurationの後に | display set を使用すると、Junosの表示がIOSと同様のスタイルに切り替わります。残念ながら、IOSには、Junosの階層表示と互換性のある表示機能はありません。

これまでの内容は基本的な概念であり、これを踏まえて、以降の章の学習を進めていきます。実際に、ジュニパーネットワークス製デバイスを開き、ユーザーアカウントを設定して、ソフトウェアをアップグレードしました。今度は、基本設定を一通り実行していきましょう。



# 第2章

## 基本設定

シングルエリアOSPF .....	18
外部BGP .....	33
内部BGP .....	43
VLAN設定 .....	51
シンプルなNAT .....	58
結論 .....	64

本書は *Day One* シリーズの 1 冊であり、実践的に学習を進めます。また、この章では、取り上げる内容を絞り込みます。具体的には、実際のネットワークで役に立つ設定を紹介します。個々の例では、特定の技術について詳しく見ていきます。さらに、例に続くまとめでは、IOS の設定と Junos の設定の相違点を比較して結論を導き出します。それぞれの設定を参考に、IOS から Junos への移行についての学習、一連の手順の比較、対応するコマンドの確認、出力内容の確認といった作業を進めます。

## シングルエリア OSPF

3 台のルーターで構成される小規模なネットワークを構築して、ルーティング情報をルーター間で交換するため、OSPF 設定を適用しましょう。図 2.1 は、今回の構成を示しています。3 台のルーターで構成されるシンプルなトポロジーであり、シングルエリア OSPF ネットワークとして設定されています。

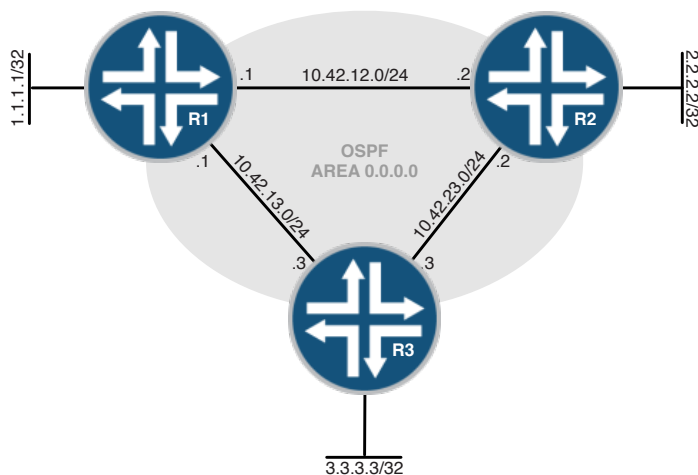


図 2.1 シングルエリア OSPF トポロジー

このセクションで実行するタスクのリストは以下のとおりです。

- R1、R2、およびR3の間にOSPFを設定する。
- すべてのルーターでループバックインタフェースをOSPFにアドバタイズして、ループバックをパッシブに設定する。
- OSPFのルーターIDを手動で設定する。

## IOS の設定

最初に、以下のステップを実行して、このネットワークを IOS で設定しましょう。

### IOS ルーターの初期接続を設定するには

ルーターを通信可能にするため、接続インタフェースで IP アドレスを設定する必要があります。以下に示すのは、IOS ルーターの IP アドレスを設定する場合に必要なステップです。

1. R1 のインタフェースの IP アドレスを設定します。

```
R1# configure terminal
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/0
R1(config-if)# ip address 10.42.13.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/1
R1(config-if)# ip address 10.42.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

2. R2 のインタフェースの IP アドレスを設定します。

```
R2# configure terminal
R2(config)# interface loopback 0
R2(config-if)# ip address 2.2.2.2 255.255.255.255
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/0
R2(config-if)# ip address 10.42.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/1
R2(config-if)# ip address 10.42.23.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# end
R2#
```

3. R3 のインタフェースの IP アドレスを設定します。

```
R3# configure terminal
R3(config)# interface loopback 0
R3(config-if)# ip address 3.3.3.3 255.255.255.255
R3(config-if)# no shutdown
R3(config-if)# interface FastEthernet 0/0
R3(config-if)# ip address 10.42.13.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface FastEthernet 0/1
R3(config-if)# ip address 10.42.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# end
R3#
```

### IOS ルーターの初期接続を確認するには

IP アドレスが正しく設定されているかどうか確認するには、シンプルな ping テストを実行して、リンクのリモートエンドが到達可能でアクティブな状態であるかどうか確認します。

1. R1 から R2 への ping を実行します。

```
R1#ping 10.42.12.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.12.2, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R1#
```

2. R1 から R3 への ping を実行します。

```
R1#ping 10.42.13.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.13.3, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R1#
```

3. R2 から R3 への ping を実行します。

```
R2#ping 10.42.23.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.0.3, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R2#
```

### IOS ルーターを OSPF 用に設定するには

ネットワークの完全な到達性を実現するには、各ルーターがネットワークを認識する必要があります。このため、通常は、OSPF (Open Shortest Path First) など、いずれかの IGP (Interior Gateway Protocol) を使用することになります。

OSPF は、LSA (Link-State Advertisement) 形式で更新情報を送出します。この情報に基づいて、エリア内の全ルーターがそれぞれのネットワーク図を作成します。

さらに詳しくは OSPF の設定方法の詳細については、『*Day One : Advanced OSPF In The Enterprise*』を参照してください (<http://www.juniper.net/dayone> で入手可能)。

1. OSPF プロセスを設定して、インタフェースを R1 のエリア 0 に割り当てます。

```
R1# configure terminal  
R1(config)# router ospf 1
```

```
R1(config-router)# network 10.42.12.1 0.0.0.0 area 0
R1(config-router)# network 10.42.13.1 0.0.0.0 area 0
R1(config-router)# end
R1#
```

2. OSPF プロセスを設定して、インタフェースを R2 のエリア 0 に割り当てます。

```
R2# configure terminal
R2(config)# router ospf 1
R2(config-router)# network 10.42.12.2 0.0.0.0 area 0
R2(config-router)# network 10.42.23.2 0.0.0.0 area 0
R2(config-router)# end
R2#
```

3. OSPF プロセスを設定して、インタフェースを R3 のエリア 0 に割り当てます。

```
R3# configure terminal
R3(config)# router ospf 1
R3(config-router)# network 10.42.13.3 0.0.0.0 area 0
R3(config-router)# network 10.42.23.3 0.0.0.0 area 0
R3(config-router)# end
R3#
```

### OSPF の設定を確認するには

OSPF が正しく設定されているかどうか確認するには、OSPF 隣接関係が正しく形成されていることをチェックします。アクティブな OSPF 隣接機器は、状態が FULL として表示されます。

1. R1 の OSPF 隣接関係を確認します。

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/BDR	00:00:32	10.42.12.2	FastEthernet0/1
3.3.3.3	1	FULL/DR	00:00:34	10.42.13.3	FastEthernet0/0

```
R1#
```

2. R2 の OSPF 隣接関係を確認します。

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/DROTHER	00:00:30	10.42.12.1	FastEthernet0/0
3.3.3.3	1	FULL/BDR	00:00:33	10.42.23.3	FastEthernet0/1

```
R2#
```

3. R3 の OSPF 隣接関係を確認します。

```
R3#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/DROTHER	00:00:37	10.42.13.1	FastEthernet0/0
2.2.2.2	1	FULL/BDR	00:00:35	10.42.23.2	FastEthernet0/1

```
R3#
```

### ループバックインタフェースを OSPF にアドバタイズしてパッシブにするには

ループバックインタフェースを OSPF にアドバタイズするのは、オプションのステップです。本書では、OSPF ドメインが適切に機能していることを簡単に確認するため実行します。実稼働ネットワークでは、ループバックインタフェースは多くの場合、OSPF ネットワークにアドバタイズされ、デバイスのルーターIDが到達可能アドレスになります。

`passive-interface` コマンドを追加すると、リンクを Type-1 LSA としてアドバタイズするようソフトウェアに通知することになりますが、そのインタフェースで隣接関係は形成されません。

1. R1 の OSPF を設定してループバックインタフェースを含めます。このループバックを経由する OSPF 隣接機器は存在しないので、`passive-interface` コマンドを追加します。

```
R1# configure terminal
R1(config)# router ospf 1
R1(config-router)# network 1.1.1.1 0.0.0.0 area 0
R1(config-router)# passive-interface lo0
R1(config-router)# end
R1#
```

2. 次に、R2 の OSPF のループバックインタフェースをパッシブに設定します。

```
R2# configure terminal
R2(config)# router ospf 1
R2(config-router)# network 2.2.2.2 0.0.0.0 area 0
R2(config-router)# passive-interface lo0
R2(config-router)# end
R2#
```

3. 最後に、R3 の OSPF のループバックインタフェースをパッシブに設定します。

```
R3# configure terminal
R3(config)# router ospf 1
R3(config-router)# network 3.3.3.3 0.0.0.0 area 0
R3(config-router)# passive-interface lo0
R3(config-router)# end
R3#
```

### OSPF のループバックインタフェースを確認するには

ループバックインタフェースを確認するには、ループバックインタフェースが実際に OSPF を実行していることをチェックしてから、他のルーターの RIB をチェックして、そのデバイスでループバックインタフェースが学習されたことを確認します。

または、エリア内のいずれかのルーターの LSDB をチェックして、同じ情報を確認します。

1. R1 (1.1.1.1/32) のループバックが OSPF 用に設定され、R2 と R3 で学習されていることを確認します。

```
R1#show ip ospf interface lo0
Loopback0 is up, line protocol is up
  Internet Address 1.1.1.1/32, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type LOOPBACK, Cost:1
  Loopback interface is treated as a stub Host
R1#
```

```
R2#show ip route 1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 10.42.12.1 on FastEthernet0/0, 00:02:41 ago
  Routing Descriptor Blocks:
  * 10.42.12.1, from 1.1.1.1, 00:02:41 ago, via FastEthernet0/0
    Route metric is 11, traffic share count is 1
R2#
```

```
R3#show ip route 1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 10.42.13.1 on FastEthernet0/0, 00:03:12 ago
  Routing Descriptor Blocks:
  * 10.42.13.1, from 1.1.1.1, 00:03:12 ago, via FastEthernet0/0
    Route metric is 11, traffic share count is 1
R3#
```

2. R2 (2.2.2.2/32) のループバックが OSPF 用に設定され、R1 と R3 で学習されていることを確認します。

```
R2#show ip ospf interface lo0
Loopback0 is up, line protocol is up
  Internet Address 2.2.2.2/32, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type LOOPBACK, Cost:1
  Loopback interface is treated as a stub Host
R2#
```

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 10.42.12.2 on FastEthernet0/1, 00:01:58 ago
  Routing Descriptor Blocks:
  * 10.42.12.2, from 2.2.2.2, 00:01:58 ago, via FastEthernet0/1
    Route metric is 11, traffic share count is 1
R1#
```

```
R3#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 10.42.23.2 on FastEthernet0/1, 00:02:16 ago
  Routing Descriptor Blocks:
  * 10.42.23.2, from 2.2.2.2, 00:02:16 ago, via FastEthernet0/1
R3#
```

```
Route metric is 11, traffic share count is 1
```

```
R3#
```

3.R3 (3.3.3.3/32) のループバックが OSPF 用に設定され、R1 と R2 で学習されていることを確認します。

```
R3#show ip ospf int lo0
```

```
Loopback0 is up, line protocol is up
Internet Address 3.3.3.3/32, Area 0
Process ID 1, Router ID 3.3.3.3, Network Type LOOPBACK, Cost:1
Loopback interface is treated as a stub Host
```

```
R3#
```

```
R1#show ip route 3.3.3.3
```

```
Routing entry for 3.3.3.3/32
Known via "ospf 1", distance 110, metric 11, type intra area
Last update from 10.42.13.3 on FastEthernet0/0, 00:04:36 ago
Routing Descriptor Blocks:
* 10.42.13.3, from 3.3.3.3, 00:04:36 ago, via FastEthernet0/0
Route metric is 11, traffic share count is 1
```

```
R1#
```

```
R2#show ip route 3.3.3.3
```

```
Routing entry for 3.3.3.3/32
Known via "ospf 1", distance 110, metric 11, type intra area
Last update from 10.42.23.3 on FastEthernet0/1, 00:04:50 ago
Routing Descriptor Blocks:
* 10.42.23.3, from 3.3.3.3, 00:04:50 ago, via FastEthernet0/1
Route metric is 11, traffic share count is 1
```

```
R2#
```

### OSPF のルーター ID を手動で設定するには

場合によっては、ルーター ID を手動で設定することにメリットがあります。この例としては、ルーターのインタフェースに割り当てられた IP アドレスと異なるルーター ID が必要な場合が挙げられます。システムでルーター ID を任意に選択可能にする場合とは対照的に、OSPF LSDB (Link-State DataBase) に確定的なエントリが必要な場合もあります。

注 OSPF のルーター ID は、ドット区切りの 10 進数形式 (IP アドレスの表記法と類似) で表現される 32 ビット値です。ただし、OSPF のエリア ID と同様に、ルーター ID は IP アドレスである必要はありませんが、多くの場合、デバイスの IP アドレスと一致するように設定されています。

1. show コマンドを使用して、R1 の現在のルーター ID を表示します。次に、ルーター ID 11.11.11.11 を使用するよう、R1 を設定します。



```
R1#show ip protocols | include ID
  Router ID 1.1.1.1
R1#configure terminal
R1(config)#router ospf 1
R1(config-router)#router-id 11.11.11.11
Reload or use "clear ip ospf process" command, for this to take effect
R1(config-router)#end
R1#clear ip ospf process
Reset ALL OSPF processes?[no]:yes
R1#
```

2. show コマンドを使用して、R2 の現在のルーター ID を表示します。  
次に、ルーター ID 22.22.22.22 を使用するよう、R2 を設定します。

```
R2#show ip protocols | include ID
  Router ID 2.2.2.2
R2#configure terminal
R2(config)#router ospf 1
R2(config-router)#router-id 22.22.22.22
Reload or use "clear ip ospf process" command, for this to take effect
R2(config-router)#end
R2#clear ip ospf process
Reset ALL OSPF processes?[no]:yes
R2#
```

3. show コマンドを使用して、R3 の現在のルーター ID を表示します。  
次に、ルーター ID 33.33.33.33 を使用するよう、R3 を設定します。

```
R3#show ip protocols | include ID
  Router ID 3.3.3.3
R3#configure terminal
R3(config)#router ospf 1
R3(config-router)#router-id 33.33.33.33
Reload or use "clear ip ospf process" command, for this to take effect
R3(config-router)#end
R3#clear ip ospf process
Reset ALL OSPF processes?[no]:yes
R3#
```

### 手動で割り当てた OSPF のルーター ID を確認するには

ルーター ID が正しく割り当てられているかどうか確認するには、show ip protocol の出力をチェックしてから、隣接するルーター ID が変更されたことをチェックして、それが正しいことを確認します。

1. show コマンドを使用して、R1 のルーター ID を確認します。

```
R1#show ip protocols | include ID
  Router ID 11.11.11.11
R1#
R1#show ip ospf neighbor
```

```

Neighbor ID      Pri   State           Dead Time   Address      Interface
33.33.33.33      1     FULL/BDR        00:00:33   10.42.13.3   FastEthernet0/0
22.22.22.22      1     FULL/BDR        00:00:36   10.42.12.2   FastEthernet0/1
R1#

```

2. show コマンドを使用して、R2 のルーター ID を確認します。

```

R2#show ip protocols | include ID
  Router ID 22.22.22.22
R2#show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address      Interface
33.33.33.33      1     FULL/BDR        00:00:31   10.42.23.3   FastEthernet0/1
11.11.11.11      1     FULL/DR         00:00:39   10.42.12.1   FastEthernet0/0
R2#

```

3. show コマンドを使用して、R3 のルーター ID を確認します。

```

R3#show ip protocols | include ID
  Router ID 33.33.33.33
R3#show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address      Interface
22.22.22.22      1     FULL/DR         00:00:30   10.42.23.2   FastEthernet0/1
11.11.11.11      1     FULL/DR         00:00:36   10.42.13.1   FastEthernet0/0
R3#

```

## Junos の設定

今回は、同じトポロジーを Junos で設定しましょう。設定の論理構造と、インタフェースおよびプロトコルの設定の配置に注意してください。このように標準化されたシンプルな構造にメリットがあることは、一見しただけではっきりとわかるでしょう。

### Junos ルーターの初期接続を設定するには

ここでは、第1章で身に付けた知識を生かして、ループバックインタフェースを含む、設定対象の各ルーターにインタフェースのアドレスを適用します。

1. R1 のインタフェースの IP アドレスを設定します。

```

cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set interfaces fe-0/0/0.0 family inet address 10.42.12.1/24

[edit]
cjones@R1# set interfaces fe-0/0/1.0 family inet address 10.42.13.1/24

[edit]
cjones@R1# set interfaces lo0.0 family inet address 1.1.1.1/32

```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. R2 のインタフェースの IP アドレスを設定します。

```
[edit]
cjones@R2# set interfaces fe-0/0/0.0 family inet address 10.42.23.2/24
```

```
[edit]
cjones@R2# set interfaces fe-0/0/1.0 family inet address 10.42.12.2/24
```

```
[edit]
cjones@R2# set interfaces lo0.0 family inet address 2.2.2.2/32
```

```
[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

3. R3 のインタフェースの IP アドレスを設定します。

```
[edit]
cjones@R3# set interfaces fe-0/0/0.0 family inet address 10.42.13.3/24
```

```
[edit]
cjones@R3# set interfaces fe-0/0/1.0 family inet address 10.42.23.3/24
```

```
[edit]
cjones@R3# set interfaces lo0.0 family inet address 3.3.3.3/32
```

```
[edit]
cjones@R3# commit and-quit
commit complete
Exiting configuration mode
```

### *Junos ルーターの初期接続を確認するには*

1. R1 から R2 への ping を実行します。

```
cjones@R1> ping 10.42.12.2 rapid
PING 10.42.12.2 (10.42.12.2):56 data bytes
!!!!
--- 10.42.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.238/2.791/4.261/0.752 ms
```

2. R1 から R3 への ping を実行します。

```
cjones@R1> ping 10.42.13.3 rapid
PING 10.42.13.3 (10.42.13.3):56 data bytes
!!!!
--- 10.42.13.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.332/5.504/17.629/6.063 ms
```

3. 最後に、R2 から R3 への ping を実行します。

```
cjones@R1> ping 10.42.23.3 rapid
PING 10.42.23.3 (10.42.23.3):56 data bytes
!!!!
--- 10.42.23.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.042/7.932/30.717/11.394 ms
```

### Junos ルーターを OSPF 用に設定するには

必要なコマンドは、各インタフェースを OSPF 階層に追加するコマンドだけなので、Junos での OSPF の設定はそれほど面倒ではありません。Junos では、OSPF を実行するインタフェースは、プロトコルのスタンプで定義します。したがって、IOS で使用される旧来のネットワークコマンドよりも、大幅に簡素化されています。

1. OSPF プロセスを設定して、インタフェースを R1 のエリア 0 に割り当てます。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set protocols ospf area 0 interface fe-0/0/0.0

[edit]
cjones@R1# set protocols ospf area 0 interface fe-0/0/1.0

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. OSPF プロセスを設定して、インタフェースを R2 のエリア 0 に割り当てます。

```
cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# set protocols ospf area 0 interface fe-0/0/0.0

[edit]
cjones@R2# set protocols ospf area 0 interface fe-0/0/1.0

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

3. OSPF プロセスを設定して、インタフェースを R3 のエリア 0 に割り当てます。

```
cjones@R3> configure
Entering configuration mode
```

```
[edit]
cjones@R3# set protocols ospf area 0 interface fe-0/0/0.0
```

```
[edit]
cjones@R3# set protocols ospf area 0 interface fe-0/0/1.0
```

```
[edit]
cjones@R3# commit and-quit
commit complete
Exiting configuration mode
```

### OSPF の設定を確認するには

OSPF が正しく設定されていることを確認するには、予期される隣接関係が存在していることを確認する方法が最も簡単です。この出力自体は、IOS の同等のコマンドの出力とよく似ているので、特に違和感はないと思われます。

1. R1 の OSPF 隣接関係を確認します。

```
cjones@R1> show ospf neighbor
Address      Interface      State      ID          Pri  Dead
10.42.12.2   fe-0/0/0.0    Full      2.2.2.2    128  31
10.42.13.3   fe-0/0/1.0    Full      3.3.3.3    128  34
```

2. R2 の OSPF 隣接関係を確認します。

```
cjones@R2> show ospf neighbor
Address      Interface      State      ID          Pri  Dead
10.42.23.3   fe-0/0/0.0    Full      3.3.3.3    128  38
10.42.12.1   fe-0/0/1.0    Full      1.1.1.1    128  39
```

3. R3 の OSPF 隣接関係を確認します。

```
cjones@R3> show ospf neighbor
Address      Interface      State      ID          Pri  Dead
10.42.13.1   fe-0/0/0.0    Full      1.1.1.1    128  39
10.42.23.2   fe-0/0/1.0    Full      2.2.2.2    128  34
```

### ループバックインタフェースを OSPF にアドバタイズしてパッシブにするには

1. R1 の OSPF を設定してループバックインタフェースを含めます。このループバックを経由する OSPF 隣接機器は存在しないので、passive ステートメントの追加はオプションのステップです。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
cjones@R1# set protocols ospf area 0 interface lo0.0 passive
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. 次に、R2 の OSPF のループバックインタフェースをパッシブに設定します。

```
cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# set protocols ospf area 0 interface lo0.0 passive

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

3. 最後に、R3 の OSPF のループバックインタフェースをパッシブに設定します。

```
cjones@R3> configure
Entering configuration mode

[edit]
cjones@R3# set protocols ospf area 0 interface lo0.0 passive

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

### OSPF のループバックインタフェースを確認するには

Junos で OSPF のループバックのアドバタイズメントを確認するには、ループバックインタフェースの OSPF ステータスをチェックするか、エリア内の他のルーターのルーティングテーブルをチェックします。

本書で Junos のルーティングテーブルを確認するのは、今回が初めてになります。Junos には、クラスフルネットワークという概念は存在しないので、IOS で見られるような "x.x.x.x/xx is subnetted" というステートメントは存在しないことに注意してください。

1. R1 (1.1.1.1/32) のループバックが OSPF 用に設定され、R2 と R3 で学習されていることを確認します。

```
cjones@R1> show ospf interface brief | match lo0.0
lo0.0          DRother 0.0.0.0          0.0.0.0          0.0.0.0          0
```

```
cjones@R2> show route protocol ospf terse 1.1.1.1
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 1.1.1.1/32	0 10	1		>10.42.12.1	

```
cjones@R3> show route protocol ospf terse 1.1.1.1
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 1.1.1.1/32	0 10	1		>10.42.13.1	

2.R2 (2.2.2.2/32) のループバックが OSPF 用に設定され、R1 と R3 で学習されていることを確認します。

```
cjones@R2> show ospf interface brief | match lo0.0
```

lo0.0	DRother	0.0.0.0	0.0.0.0	0.0.0.0	0
-------	---------	---------	---------	---------	---

```
cjones@R1> show route protocol ospf terse 2.2.2.2
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 2.2.2.2/32	0 10	1		>10.42.12.2	

```
cjones@R3> show route protocol ospf terse 2.2.2.2
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 2.2.2.2/32	0 10	1		>10.42.23.2	

3.R3 (3.3.3.3/32) のループバックが OSPF 用に設定され、R1 と R2 で学習されていることを確認します。

```
cjones@R3> show ospf interface brief | match lo0.0
```

lo0.0	DRother	0.0.0.0	0.0.0.0	0.0.0.0	0
-------	---------	---------	---------	---------	---

```
cjones@R1> show route protocol ospf terse 3.3.3.3
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 3.3.3.3/32	0 10	1		>10.42.13.3	

```
cjones@R2> show route protocol ospf terse 3.3.3.3
```

```
inet.0:9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

A Destination	P Prf	Metric 1	Metric 2	Next hop	AS path
* 3.3.3.3/32	0 10	1		>10.42.23.3	

### OSPF のルーター ID を手動で設定するには

Junos ソフトウェアは、rpd の開始時（通常は起動時）に martian アドレス以外のアドレスが付与された最初のインタフェースからルーター ID を割り当てます。この場合、ほぼ必ず、ループバックインタフェースに割り当てられたアドレスがルーターで選択されて、ルーター ID として使用されます。この処理は、IOS の場合と基本的に同じです。

1. show コマンドを使用して、R1 の現在のルーター ID を表示します。次に、ルーター ID 11.11.11.11 を使用するよう、R1 を設定します。

```
cjones@R1> show ospf overview | match "Router ID"
Router ID:1.1.1.1
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options router-id 11.11.11.11

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. show コマンドを使用して、R2 の現在のルーター ID を表示します。次に、ルーター ID 22.22.22.22 を使用するよう、R2 を設定します。

```
cjones@R2> show ospf overview | match "Router ID"
Router ID:2.2.2.2

cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# set routing-options router-id 22.22.22.22

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

3. show コマンドを使用して、R3 の現在のルーター ID を表示します。次に、ルーター ID 33.33.33.33 を使用するよう、R3 を設定します。

```
cjones@R3> show ospf overview | match "Router ID"
Router ID:3.3.3.3

cjones@R3> configure
Entering configuration mode

[edit]
cjones@R3# set routing-options router-id 33.33.33.33

[edit]
cjones@R3# commit and-quit
```



```
commit complete
Exiting configuration mode
```

### 手動で割り当てた OSPF のルーター ID を確認するには

1. show コマンドを使用して、R1 のルーター ID を確認します。

```
cjones@R1> show ospf overview | match "Router ID"
Router ID:11.11.11.11
```

2. show コマンドを使用して、R2 のルーター ID を確認します。

```
cjones@R2> show ospf overview | match "Router ID"
Router ID:22.22.22.22
```

3. show コマンドを使用して、R3 のルーター ID を確認します。

```
cjones@R3> show ospf overview | match "Router ID"
Router ID:33.33.33.33
```

### まとめ

このセクションのタスクは、シンプルな OSPF ネットワークの設定でした。ただし、IOS では、旧来のネットワークコマンドで OSPF の実行インタフェースを記述していますが、Junos では対照的に、Junos のプロトコル階層に従って管理者が OSPF の実行インタフェースを指定できるようにすることで、どのようにプロセスを簡素化したかという点を説明しました。繰り返しになりますが、このような細かい差異であっても、ネットワークが大規模になれば相応に大きくなります。

## 外部 BGP

次の例では、2 つの AS (Autonomous System) 間にシンプルな外部 BGP のピアリングセッションを設定しましょう。2 つの OS を比較することで、Junos の方法論だけでなく、その一貫性のあるポリシー設定のフレームワークについても学習できているはずです。

直接接続している 2 台のルーターを使用します。EBGP ピアリングでは、物理インタフェースを使用します。以下に示すのは、トポロジー図です。

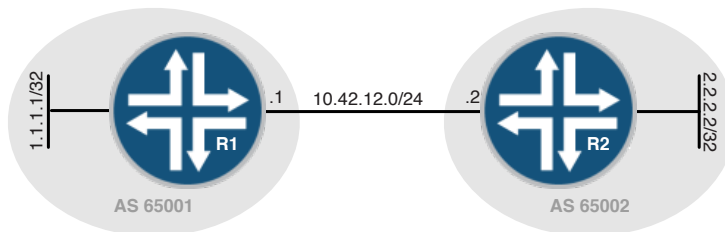


図 2.2 外部 BGP トポロジー

このセクションで実行するタスクのリストは以下の通りです。

- AS 65001のR1とAS 65002のR2の間にEBGPピアリングを設定する。
- ピアに対する物理インタフェースを使用する。
- ループバックインタフェースを隣接機器にアドバタイズする。

## IOS の設定

最初に、IOS を使用して、このネットワークを設定しましょう。

### IOS ルーターの初期接続を設定するには

1. R1 のインタフェースの IP アドレスを設定します。

```
R1# configure terminal
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/0
R1(config-if)# ip address 10.42.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

2. R2 のインタフェースの IP アドレスを設定します。

```
R2# configure terminal
R2(config)# interface loopback 0
R2(config-if)# ip address 2.2.2.2 255.255.255.255
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/0
R2(config-if)# ip address 10.42.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# end
R2#
```

### 初期接続を確認するには

1. R1 から R2 への ping を実行します。

```
R1#ping 10.42.12.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.42.12.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

### 物理インタフェースを使用して EBGP ピアリングを設定するには

1. R1 の EBGP を AS 65002 の R2 のピアに設定します。

```
R1#configure terminal
R1(config)#router bgp 65001
```

```
R1(config-router)# no synchronization
R1(config-router)# neighbor 10.42.12.2 remote-as 65002
R1(config-router)# no auto-summary
R1(config-router)#end
R1#
```

2. R2 の EBGP を AS 65001 の R1 のピアに設定します。

```
R2#configure terminal
R2(config)#router bgp 65002
R2(config-router)# no synchronization
R2(config-router)# neighbor 10.42.12.1 remote-as 65001
R2(config-router)# no auto-summary
R2(config-router)#end
R2#
```

### EBGP ピアリングを確認するには

1. R1 の EBGP ピアリングが Established 状態であることを確認します。

```
R1#show ip bgp neighbors
BGP neighbor is 10.42.12.2, remote AS 65002, external link
  BGP version 4, remote router ID 2.2.2.2
  BGP state = Established, up for 00:00:30
  Last read 00:00:30, last write 00:00:30, hold time is 180, keepalive interval is 60
seconds
...
```

2. R2 の EBGP ピアリングが Established 状態であることを確認します。

```
R2#show ip bgp neighbors
BGP neighbor is 10.42.12.1, remote AS 65001, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:01:57
  Last read 00:00:57, last write 00:00:57, hold time is 180, keepalive interval is 60
seconds
...
```

### ループバックアドレスをアドバタイズするよう EBGP を設定するには

1. ループバックアドレスをアドバタイズするよう R1 の BGP プロセスを設定します。

```
R1#configure terminal
R1(config)#router bgp 65001
R1(config-router)#network 1.1.1.1 mask 255.255.255.255
R1(config-router)#end
R1#
```

2. ループバックアドレスをアドバタイズするよう R2 の BGP プロセスを設定します。

```
R2#configure terminal
R2(config)#router bgp 65002
R2(config-router)#network 2.2.2.2 mask 255.255.255.255
R2(config-router)#end
R2#
```

### EBGP がループバックアドレスをアドバタイズしていることを確認するには

1. R2 の BGP 隣接機器のサマリーを表示して、BGP プレフィックスが受信されているかどうか判断します。

```
R2#show ip bgp summary | include Nei|65001
Neighbor      V   AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down   State/PfxRcd
10.42.12.1    4 65001      6      6       3    0    0 00:02:24      1
```

2. R2 の受信済みルートを表示します。

```
R2#show ip bgp regexp ^65001
BGP table version is 3, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ?- incomplete
```

```
Network          Next Hop          Metric LocPrf Weight Path
*> 1.1.1.1/32     10.42.12.1        0             0 65001 i
```

3. R2 の RIB を表示します。

```
R2#show ip route bgp | e subnetted
B          1.1.1.1 [20/0] via 10.42.12.1, 00:07:17
```

4. R1 の BGP 隣接機器のサマリーを表示します。

```
R1#show ip bgp summ | include Nei|65002
Neighbor      V   AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down   State/PfxRcd
10.42.12.2    4 65002      9      9       3    0    0 00:05:16      1
```

5. R1 の受信済みルートを表示します。

```
R1#show ip bgp regexp ^65002
BGP table version is 3, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ?- incomplete
```

```
Network          Next Hop          Metric LocPrf Weight Path
*> 2.2.2.2/32     10.42.12.2        0             0 65002 i
```

6. R1 の RIB を表示します。

```
R1#show ip route bgp | exclude subnetted
B          2.2.2.2 [20/0] via 10.42.12.2, 00:05:30
```

## Junos の設定

今回は、Junos ルーターで同じネットワークを設定して、その相違を確認しましょう。

### Junos ルーターの初期接続を設定するには

1. R1 のインタフェースの IP アドレスを設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set interfaces fe-0/0/0.0 family inet address 10.42.12.1/24

[edit]
cjones@R1# set interfaces lo0.0 family inet address 1.1.1.1/32
```

2. R2 のインタフェースの IP アドレスを設定します。

```
[edit]
cjones@R1# set interfaces fe-0/0/0.0 family inet address 10.42.12.2/24

[edit]
cjones@R1# set interfaces lo0.0 family inet address 2.2.2.2/32
```

3. 設定をコミットします。

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

4. R1 のインタフェース設定を確認します。

```
cjones@R1> show configuration interfaces
fe-0/0/0 {
  unit 0 {
    family inet {
      address 10.42.12.1/24;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

5. R2 のインタフェース設定を確認します。

```
cjones@R2> show configuration interfaces
fe-0/0/0 {
  unit 0 {
    family inet {
      address 10.42.12.2/24;
    }
  }
}
lo0 {
  unit 0 {
```

```

        family inet {
            address 2.2.2.2/32;
        }
    }
}

```

6. R1 から R2 に ping を実行して、初期接続を確認します。

```

cjones@R1> ping 10.42.12.2 rapid
PING 10.42.12.2 (10.42.12.2):56 data bytes
!!!!
--- 10.42.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.238/2.791/4.261/0.752 ms

```

### 物理インタフェースを使用してEBGPピアリングを設定するには

Junos の BGP 設定を IOS と比較した場合、最も大きな違いとして、Junos では、(AS 番号を除く) BGP 隣接機器の設定は通常、コマンド1つで実行できることが挙げられます。

1. R1 の EBGP を AS 65002 の R2 のピアに設定します。

```

[edit]
cjones@R1# set routing-options autonomous-system 65001

```

```

[edit]
cjones@R1# set protocols bgp group EBGP neighbor 10.42.12.2 peer-as 65002

```

2. R2 の EBGP を AS 65001 の R1 のピアに設定します。

```

[edit]
cjones@R1# set routing-options autonomous-system 65002

```

```

[edit]
cjones@R1# set protocols bgp group EBGP neighbor 10.42.12.1 peer-as 65001

```

3. 設定をコミットします。

```

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode

```

4. R1 の設定を確認します。

```

cjones@R1> show configuration routing-options
autonomous-system 65001;

```

```

cjones@R1> show configuration protocols bgp
group EBGP {
    neighbor 10.42.12.2 {
        peer-as 65002;
    }
}

```

5. R2 の設定を確認します。

```
cjones@R2> show configuration routing-options
autonomous-system 65002;
```

```
cjones@R2> show configuration protocols bgp
group EBGP {
  neighbor 10.42.12.1 {
    peer-as 65001;
  }
}
```

物理インタフェースを使用してEBGPピアリングを確認するには

1. 最初に、R1 のEBGPピアリングがEstablished状態であることを確認します。

```
cjones@R1> show bgp neighbor 10.42.12.2 | match Established
Type:External State:Established Flags:<ImportEval Sync>
```

2. 次に、R2 のEBGPピアリングがEstablished状態であることを確認します。

```
cjones@R2> show bgp neighbor 10.42.12.1 | match Established
Type:External State:Established Flags:<ImportEval Sync>
```

ループバックアドレスをアドバタイズするようEBGPを設定するには

BGPセッションがEstablished状態になった時点で、設定をテストする必要があります。BGPが動作していることを適切に確認するには、いくつかのプレフィックスをBGP隣接機器にアドバタイズする必要があります。実際に、やってみましょう。

具体的には、Junosのほとんどのルーティング操作と同様に、ポリシーを使用します。

1. R1のループバックインタフェースと適合するポリシーを作成して、BGPプロセスに適用できるようにします。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
```

```
cjones@R1# edit policy-options policy-statement EBGP_ADVERTISE_LOOPBACK
```

```
[edit policy-options policy-statement EBGP_ADVERTISE_LOOPBACK]
```

```
cjones@R1# set term ADVERTISE_L00 from protocol direct
```

```
[edit policy-options policy-statement EBGP_ADVERTISE_LOOPBACK]
```

```
cjones@R1# set term ADVERTISE_L00 from route-filter 1.1.1.1/32 exact
```

```
[edit policy-options policy-statement EBGP_ADVERTISE_LOOPBACK]
```

```
cjones@R1# set term ADVERTISE_L00 then accept
```

```
[edit policy-options policy-statement EBGP_ADVERTISE_LOOPBACK]
```

```
cjones@R1# top
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. R1 のポリシー設定を確認します。

```
cjones@R1> show configuration policy-options
policy-statement EBG_P_ADVERTISE_LOOPBACK {
  term ADVERTISE_L00 {
    from {
      protocol direct;
      route-filter 1.1.1.1/32 exact;
    }
    then accept;
  }
}
```

3. R1 の BGP 隣接機器でポリシーをエクスポートポリシーとして設定します。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
cjones@R1# set protocols bgp group EBG_P neighbor 10.42.12.2 export EBG_P_ADVERTISE_LOOPBACK
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

4. R1 の BGP エクスポートポリシー設定を確認します。

```
cjones@R1> show configuration protocols bgp
group EBG_P {
  neighbor 10.42.12.2 {
    export EBG_P_ADVERTISE_LOOPBACK;
    peer-as 65002;
  }
}
```

5. R2 のループバックインタフェースと適合するポリシーを作成して、BGP プロセスに適用できるようにします。

```
[edit]
cjones@R2# edit policy-options policy-statement EBG_P_ADVERTISE_LOOPBACK
```

```
[edit policy-options policy-statement EBG_P_ADVERTISE_LOOPBACK]
cjones@R2# set term ADVERTISE_L00 from protocol direct
```

```
[edit policy-options policy-statement EBG_P_ADVERTISE_LOOPBACK]
cjones@R2# set term ADVERTISE_L00 from route-filter 1.1.1.1/32 exact
```

```
[edit policy-options policy-statement EBG_P_ADVERTISE_LOOPBACK]
cjones@R2# set term ADVERTISE_L00 then accept
```



```
[edit policy-options policy-statement EBG_P_ADVERTISE_LOOPBACK]
cjones@R2# top
```

```
[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

6. R2 のポリシー設定を確認します。

```
cjones@R2> show configuration policy-options
policy-statement EBG_P_ADVERTISE_LOOPBACK {
  term ADVERTISE_L00 {
    from {
      protocol direct;
      route-filter 2.2.2.2/32 exact;
    }
    then accept;
  }
}
```

7. R2 の BGP 隣接機器でポリシーをエクスポートポリシーとして設定します。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
cjones@R1# set protocols bgp group EBG_P neighbor 10.42.12.1 export EBG_P_ADVERTISE_
LOOPBACK
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

8. R2 の BGP エクスポートポリシー設定を確認します。

```
cjones@R2> show configuration protocols bgp
group EBG_P {
  neighbor 10.42.12.1 {
    export EBG_P_ADVERTISE_LOOPBACK;
    peer-as 65001;
  }
}
```

*ループバックアドレスが BGP によってアドバタイズされているかどうか確認するには*

BGP エクスポートポリシーを確認するには、BGP サマリー情報を確認して Adj-RIB-In テーブルをチェックします（このテーブルには、インポートポリシー（存在する場合）の適用前に受信された BGP プレフィックスのリストが表示されます）。ルーティングテーブルを簡単にチェックするだけでも、プレフィックスがインストールされていることの確認になります。

1. R1 の BGP 隣接機器のサマリーを表示して、BGP プレフィックスが受信されているかどうか判断します。

```
cjones@R1> show bgp summary
Groups:1 Peers:1 Down peers:0
Table          Tot Paths  Act Paths Suppressed    History Damp State   Pending
inet.0         1          1          0          0         0         0         0
Peer          AS        InPkt    OutPkt    OutQ    Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.42.12.2   65002    12        13        0        0    4:04 1/1/1/0
0/0/0/0
```

2. R1 の受信済みルートを表示します。

```
cjones@R1> show route receive-protocol bgp 10.42.12.2
inet.0:6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 2.2.2.2/32  10.42.12.2                65002 I
```

3. R1 の RIB を表示します。

```
cjones@R1> show route protocol bgp
inet.0:6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.2.2/32          *[BGP/170] 00:05:22, localpref 100
                    AS path:65002 I
                    > to 10.42.12.2 via fe-0/0/0.0
```

4. R2 の BGP 隣接機器のサマリーを表示して、BGP プレフィックスが受信されているかどうか判断します。

```
cjones@R2> show bgp summary
Groups:1 Peers:1 Down peers:0
Table          Tot Paths  Act Paths Suppressed    History Damp State   Pending
inet.0         1          1          0          0         0         0         0
Peer          AS        InPkt    OutPkt    OutQ    Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.42.12.1   65001    15        15        0        0    7:21 1/1/1/0
0/0/0/0
```

5. 最後に、R2 の受信済みルートを表示します。

```
cjones@R2> show route receive-protocol bgp 10.42.12.1
inet.0:6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 1.1.1.1/32  10.42.12.1                65001 I
```

6. R2 の RIB を表示します。

```
cjones@R2> show route protocol bgp
inet.0:6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

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

```
1.1.1.1/32      *[BGP/170] 00:16:32, localpref 100
                 AS path:65001 I
                 > to 10.42.12.1 via fe-0/0/0.0
```

まとめ

AS 番号とピアリングを設定する2つのコマンドだけでシンプルな BGP ピアリングを設定できる機能は、Junos の明らかなメリットです。IOS で BGP プロセスにルートアドバタイズメントを直接設定できるのも便利な機能ですが、このような場合には、Junos ポリシー設定が真価を発揮することに疑問の余地はありません。一貫性のあるフレームワークに基づいてポリシー設定を用意できることは、Junos が実際のネットワークに適していることを示す明らかな例です。ネットワークの拡張や、新しいサービスの必要性に応じて、このメリットも大きくなります。

もう1つ、触れておくべき点として、同期の概念が挙げられます。IOS では、ルートは BGP 経由でアナウンスされる前に、IGP のルーティングテーブルでアクティブであることが求められます。通常、この同期の概念は無効です。Junos では、このような基準はありません。

## 内部 BGP

このセクションでは、IBGP (Internal BGP: 内部 BGP) を設定しましょう。IBGP ピアリングはループバックアドレス間のピアリングを目的として設定され、リモートループバックアドレスを到達可能にするためには、スタティックルートの追加が必要になります。IOS エンジニアは、BGP のピアリングセッションをグループ化する Junos の機能に注目する必要があります。

このセクションでは、図 2.3 のトポロジー図に示すように、単一の AS 内で直接接続された2台のルーターを使用します。

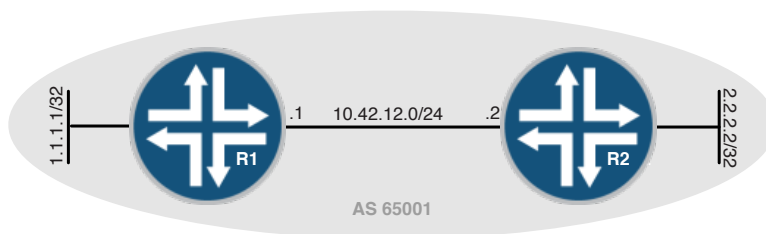


図 2.3 内部 BGP トポロジー

このセクションで実行するタスクのリストは以下のとおりです。

- ループバックアドレスへの完全な到達性を確保するため、R1とR2のスタティックルートを設定する。

- AS 65001内のR1とR2の間にIBGPピアリングを設定する。
- IBGPピアリング用のループバックを使用する。

## IOS の設定

最初に、IOS を使用して、このネットワークを設定しましょう。一連の手順は、特に目新しいものではありません。

### IOS ルーターの初期接続を設定するには

1. R1 のインタフェースの IP アドレスを設定します。

```
R1# configure terminal
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/0
R1(config-if)# ip address 10.42.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

2. R2 のインタフェースの IP アドレスを設定します。

```
R2# configure terminal
R2(config)# interface loopback 0
R2(config-if)# ip address 2.2.2.2 255.255.255.255
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/0
R2(config-if)# ip address 10.42.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# end
R2#
```

### 初期接続を確認するには

1. R1 から R2 への ping を実行します。

```
R1#ping 10.42.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.42.12.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

### フル接続用にスタティックルートを設定するには

1. R1 のスタティックルートを R2 のループバックアドレスに設定して、R1 を参照します。R2 の物理インタフェースをネクストホップに設定します。

```
R1#configure terminal
R1(config)#ip route 2.2.2.2 255.255.255.255 10.42.12.2
R1(config)#end
R1#
```

2. R1 のスタティックルートを R2 のループバックアドレスに設定して、R2 の物理インタフェースをネクストホップとして参照します。

```
R2#configure terminal
R2(config)#ip route 1.1.1.1 255.255.255.255 10.42.12.1
R2(config)#end
R2#
```

スタティックルートがアクティブな状態であり、フル接続が確立されたことを確認するには、

1. R1 の RIB を表示します。

```
R1#show ip route static | exclude subnett
S      2.2.2.2 [1/0] via 10.42.12.2
R1#
```

2. R2 の RIB を表示します。

```
R2#show ip route static | exclude subnett
S      1.1.1.1 [1/0] via 10.42.12.1
R2#
```

3. R1 から R2 のループバックへの ping を実行します。

```
R1#ping 2.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/22/24 ms
R1#
```

4. R2 から R1 のループバックへの ping を実行します。

```
R2#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/20 ms
R2#
```

ループバック間の IBGP ピアリングを設定するには

1. R1 の IBGP を設定します。

```
R1#configure terminal
R1(config)#router bgp 65001
R1(config-router)#no auto-summary
R1(config-router)#no synchronization
R1(config-router)#neighbor 2.2.2.2 remote-as 65001
R1(config-router)#neighbor 2.2.2.2 update-source lo0
R1(config-router)#end
R1#
```

2. R2 の IBGP を設定します。

```
R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)#no auto-summary
R2(config-router)#no synchronization
R2(config-router)#neighbor 1.1.1.1 remote-as 65001
R2(config-router)#neighbor 1.1.1.1 update-source lo0
R2(config-router)#end
R2#
```

*ループバック間の IBGP ピアリングを確認するには*

1. R1 の IBGP ピアリングが Established 状態であることを確認します。

```
R1#show ip bgp neighbors
BGP neighbor is 2.2.2.2, remote AS 65001, internal link
  BGP version 4, remote router ID 2.2.2.2
  BGP state = Established, up for 00:01:01
  Last read 00:00:01, last write 00:00:01, hold time is 180, keepalive interval is 60
seconds
...
```

2. R2 の IBGP ピアリングが Established 状態であることを確認します。

```
R2#show ip bgp neighbors
BGP neighbor is 1.1.1.1, remote AS 65001, internal link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:02:42
  Last read 00:00:42, last write 00:00:42, hold time is 180, keepalive interval is 60
seconds
...
```

## Junos の設定

今回は、同じ条件を Junos ルーターで設定しましょう。この設定は、前のタスクの EBGP の設定とほぼ同じです。主な違いは、タイプが internal になっていることです。また、local-address 設定オプションを使用して、IBGP メッセージをループバックアドレスから取得するよう指定していることに注意してください。

IOS エンジニアであれば、Junos では AS 番号が IBGP 隣接機器に対して設定されていないことに気付くかもしれません。これは、Junos では BGP ピアリングを internal として明示的に定義するのに対して、IOS ではローカルルーターと同じ AS でピアを設定することにより、タイプは internal として暗黙的に定義されることに起因します。

*Junos ルーターの初期接続を設定するには*

1. 最初に、R1 のインタフェースの IP アドレスを設定します。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
cjones@R1# set interfaces fe-0/0/0.0 family inet address 10.42.12.1/24
```

```
[edit]
cjones@R1# set interfaces lo0.0 family inet address 1.1.1.1/32
```

2. R2 のインタフェースの IP アドレスを設定します。

```
[edit]
cjones@R1# set interfaces fe-0/0/0.0 family inet address 10.42.12.2/24
```

```
[edit]
cjones@R1# set interfaces lo0.0 family inet address 2.2.2.2/32
```

3. 設定をコミットします。

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

4. R1 のインタフェース設定を確認します。

```
cjones@R1> show configuration interfaces
fe-0/0/0 {
  unit 0 {
    family inet {
      address 10.42.12.1/24;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

5. R2 のインタフェース設定を確認します。

```
cjones@R2> show configuration interfaces
fe-0/0/0 {
  unit 0 {
    family inet {
      address 10.42.12.2/24;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 2.2.2.2/32;
    }
  }
}
```

6. 最後に、R1 から R2 に ping を実行して、初期接続を確認します。

```
cjones@R1> ping 10.42.12.2 rapid
PING 10.42.12.2 (10.42.12.2):56 data bytes
!!!!
--- 10.42.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.238/2.791/4.261/0.752 ms
```

### フル接続用にスタティックルートを設定するには

OSPF がネットワーク上で動作し、すでに完全な到達性が確保されているので、この例では隣接機器のループバックアドレスへのスタティックルートは不要です。ただし、この例では、設定の実行に必要な構文を紹介するため、スタティックルートを設定します。

1. R1 のスタティックルートを R2 のループバックアドレスに設定して、R2 の物理インタフェースをネクストホップとして参照します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options static route 2.2.2.2/32 next-hop 10.42.12.2

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. R1 のスタティックルート設定を確認します。

```
cjones@R1> show configuration routing-options
static {
    route 2.2.2.2/32 next-hop 10.42.12.2;
}
```

3. R2 のスタティックルートを R1 のループバックアドレスに設定して、R1 の物理インタフェースをネクストホップとして参照します。

```
cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# set routing-options static route 1.1.1.1/32 next-hop 10.42.12.1

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

4. R2 のスタティックルート設定を確認します。

```
cjones@R2> show configuration routing-options
static {
    route 1.1.1.1/32 next-hop 10.42.12.1;
}
```



スタティックルートがアクティブな状態であり、フル接続が確立されたことを確認するには、

1. R1 の RIB を表示します。

```
cjones@R1> show route
```

```
inet.0:4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

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

```
1.1.1.1/32      *[Direct/0] 00:09:23
                 > via lo0.0
2.2.2.2/32      *[Static/5] 00:06:02
                 > to 10.42.12.2 via fe-0/0/0.0
10.42.12.0/24   *[Direct/0] 00:09:23
                 > via fe-0/0/0.0
10.42.12.1/32   *[Local/0] 00:09:23
                 Local via fe-0/0/0.0
```

2. R2 の RIB を表示します。

```
cjones@R2> show route
```

```
inet.0:4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

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

```
1.1.1.1/32      *[Static/5] 00:07:27
                 > to 10.42.12.1 via fe-0/0/0.0
2.2.2.2/32      *[Direct/0] 00:12:11
                 > via lo0.0
10.42.12.0/24   *[Direct/0] 00:08:16
                 > via fe-0/0/0.0
10.42.12.2/32   *[Local/0] 00:10:04
                 Local via fe-0/0/0.0
```

3. R1 から R2 のループバックへの ping を実行します。

```
cjones@R1> ping 2.2.2.2 rapid
```

```
PING 2.2.2.2 (2.2.2.2):56 data bytes
```

```
!!!!!
```

```
--- 2.2.2.2 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max/stddev = 0.244/0.370/0.575/0.117 ms
```

4. R2 から R1 のループバックへの ping を実行します。

```
cjones@R2> ping 1.1.1.1 rapid
```

```
PING 1.1.1.1 (1.1.1.1):56 data bytes
```

```
!!!!!
```

```
--- 1.1.1.1 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max/stddev = 0.374/0.449/0.510/0.045 ms
```

### ループバック間の IBGP ピアリングを設定するには

繰り返しになりますが、Junos の BGP 設定はとてもシンプルであり、AS 番号の設定と、隣接機器を設定するコマンド1つを実行するだけで済みます。

1. R1 の IBGP を設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options autonomous-system 65001

[edit]
cjones@R1# set protocols bgp group IBGP type internal neighbor 2.2.2.2 local-address 1.1.1.1

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

2. R1 の IBGP 設定を確認します。

```
cjones@R1> show configuration routing-options autonomous-system 65001;

cjones@R1> show configuration protocols bgp
group IBGP {
    type internal;
    neighbor 2.2.2.2 {
        local-address 1.1.1.1;
        peer-as 65001;
    }
}
```

3. R2 の IBGP を設定します。

```
cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# set routing-options autonomous-system 65001

[edit]
cjones@R2# set protocols bgp group IBGP type internal neighbor 1.1.1.1 local-address 2.2.2.2

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

## ループバック間の IBGP ピアリングを確認するには

IBGP を確認するには、EBGP の場合と同じ方法でピアリングセッションが Established 状態であることをチェックします。

1. R1のIBGPピアリングがEstablished状態であることを確認します。

```
cjones@R1> show bgp neighbor 2.2.2.2 | match Established
Type:Internal    State:Established    Flags:<ImportEval Sync>
```

2. R2のIBGPピアリングがEstablished状態であることを確認します。

```
cjones@R2> show bgp neighbor 1.1.1.1 | match Established
Type:Internal    State:Established    Flags:<ImportEval Sync>
```

### まとめ

Junos では、IBGP の設定は、EBGP の設定とほぼ同じ設定です。主な違いは、type internal を使用して、peer-as がローカル AS と一致することです。ただし、peer-as コマンドは不要であり、タイプが internal の場合は必ず、ピア AS とローカル AS は暗黙的に同じになります。

Junos のメリットの1つとしては、BGP ピアリングセッションをグループ化できることが挙げられます。このグループ化により、IBGP と EBGP のピアリングセッションの論理的な区分、サービスプロバイダ接続の区分、その他の考え得る目的に対応できます。

Junos では、IOS の no auto summary と同等のコマンドは不要である点にも注意してください。これは、Junos には、クラスフルネットワークやクラスレスネットワークという概念が存在しないことに起因します。CIDR は排他的に使用されます。

## VLAN 設定

ここでは、基本的なスイッチング設定のタスクを実行して、2つのOSの比較を確認しましょう。IOS エンジニアにとって重要なポイントは、Junos での VLAN の命名方法であり、VLAN-ID タグは VLAN に基づいて設定されるということです。

このセクションのトポロジは単一のレイヤー 3 スイッチで構成されているので、トポロジ図は不要です。

このセクションで実行するタスクのリストは以下のとおりです。

- VLANを設定する。
- 新しいVLANをアクセス(タグなし)インタフェースに割り当てる。
- トランク(802.1Qタグ付き)インタフェースを作成する。
- レイヤー3インタフェースを作成する。

## IOS の設定

最初に、IOS を使用して、このネットワークを設定します。タスク自体は、特に目新しいものではありません。

### IOS スイッチに VLAN を設定するには

1. VLAN を作成して、名前を割り当てます。

```
SW1#configure terminal
SW1(config)#vlan 10
SW1(config-vlan)#name IOS_VLAN
SW1(config-vlan)#exit
SW1(config)#end
SW1#
```

### 作成および命名済みの VLAN を確認するには

1. VLAN リストを表示します。

```
SW1#show vlan br
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Gi0/1, Gi0/2
10	IOS_VLAN	active	

```
SW1#
```

### VLAN をアクセスインタフェースに割り当てるには

1. インタフェースをアクセスインタフェースとして設定し、VLAN 10 をタグなしで割り当てます。

```
SW1#configure terminal
SW1(config)#interface fa0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#end
SW1#
```

### VLAN が fa0/2 にタグなしで追加されたことを確認するには

1. もう一度、VLAN リストを表示します。

```
SW1#show vlan br
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Gi0/1, Gi0/2
10 IOS_VLAN	active	<b>Fa0/2</b>

```
SW1#
```

### すべての VLAN を伝送するトランクインタフェースを作成するには

1. IOS のデフォルトでは、設定済みのすべての VLAN をトランクインタフェースで伝送します。したがって、インタフェースを 802.1Q タグ付きインタフェースとして設定しましょう。

```
SW1#configure terminal
```

```
SW1(config)#interface fa0/24
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#end
SW1#
```

### トランクインタフェースを確認するには

1. トランク設定を表示して確認します。

```
SW1#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	<b>trunking</b>	1
Port	Vlans allowed on trunk			
Fa0/24	1-4094			
Port	Vlans allowed and active in management domain			
Fa0/24	1,10			
Port	Vlans in spanning tree forwarding state and not pruned			
Fa0/24	1,10			

### レイヤー 3 インタフェース (IOS の SVI) を作成するには

1. IOS\_VLAN に L3 インタフェースを設定します。

```
SW1#configure terminal
```

```
SW1(config)#interface vlan 10
SW1(config-if)#ip address 10.10.10.1 255.255.255.0
SW1(config-if)#end
SW1#
```

### レイヤー3 インタフェースを確認するには

1. 確認の最後のプロセスとして、スイッチのインタフェース IP アドレスのリストを表示します。

```
SW1#show ip int br | include Vlan
Vlan1          unassigned      YES NVRAM  administratively down down
Vlan10        10.10.10.1      YES manual up           up
```

## Junos の設定

次に、Junos で同じタスクを実行しましょう。IOS エンジニアは、VLAN を命名して設定する方法に注目してください。

### Junos スイッチに VLAN を設定するには

1. VLAN を作成して、VLAN ID を割り当てます。

```
cjones@SW1> configure
Entering configuration mode

[edit]
cjones@SW1# set vlans JUNOS_VLAN vlan-id 10

[edit]
cjones@SW1# commit and-quit
commit complete
Exiting configuration mode
```

2. VLAN 設定を確認します。

```
cjones@SW1> show configuration vlans
JUNOS_VLAN {
    vlan-id 10;
}
```

### 作成および命名済みの VLAN を確認するには

1. VLAN リストを表示して確認します。

```
cjones@SW1> show vlans JUNOS_VLAN
Name      Tag      Interfaces
JUNOS_VLAN 10
None
```

### VLAN をアクセスインタフェースに割り当てるには

Junos では、実際に VLAN をインタフェースに割り当てるには、以下の2種類の方法があります。

- VLAN設定
- インタフェース

インタフェースのアクセス VLAN を VLAN 設定のスタンザで設定しましょう。どちらの場合でも、イーサネットスイッチングファミリは、インタフェースで設定する必要があります。

1. イーサネットスイッチングファミリのインタフェースを設定して、VLAN 10 を VLAN 設定のスタンザに割り当てます。

```
cjones@SW1> configure
Entering configuration mode

[edit]
cjones@SW1# set interfaces fe-0/0/0 unit 0 family ethernet-switching port-mode access

[edit]
cjones@SW1# set vlans JUNOS_VLAN interface fe-0/0/0.0

[edit]
cjones@SW1# commit and-quit
commit complete
Exiting configuration mode
```

2. アクセスインタフェースの設定を確認します。

```
cjones@SW1> show configuration interfaces fe-0/0/0.0
family ethernet-switching;
port-mode access;

cjones@SW1> show configuration vlans
JUNOS_VLAN {
  vlan-id 10;
  interface {
    fe-0/0/0.0;
  }
}
```

*VLAN が fe-0/0/0.0 にタグなしで追加されたことを確認するには*

1. VLAN リストを表示します。

```
cjones@SW1> show vlans JUNOS_VLAN
Name      Tag  Interfaces
JUNOS_VLAN 10   fe-0/0/0.0
```

2. VLAN リストの詳細出力を表示して、VLAN が fe-0/0/0.0 でタグなしの状態であることを確認します。

```
cjones@SW1> show vlans extensive JUNOS_VLAN
VLAN:JUNOS_VLAN, Created at:Sun Jun 24 12:54:01 2012
802.1Q Tag:10, Internal index:2, Admin State:Enabled, Origin:Static
Protocol:Port Mode, Mac aging time:300 seconds
Number of interfaces:Tagged 0 (Active = 0), Untagged 1 (Active = 0)
fe-0/0/0.0, untagged, access
```

3. イーサネットスイッチングインタフェースの詳細を表示します。

```
cjones@SW1> show ethernet-switching interfaces detail
Interface: fe-0/0/0.0 Index:69
State:UP
Vlans:JUNOS_VLAN(untagged)
```

すべての VLAN を伝送するトランクインタフェースを作成するには

今度は、VLAN にインタフェースを設定するのではなく、インタフェースに VLAN を設定しましょう。IOS エンジニアであれば、この方法に慣れているでしょう。どちらの方法を選択しても問題はありますが、重要なポイントは、一貫性のあるアプローチで対処することです。

1. イーサネットスイッチングファミリのインタフェースを設定して、このインタフェースの VLAN に JUNOS\_VLAN タグを追加します。

```
cjones@SW1> configure
Entering configuration mode

[edit]
cjones@SW1# set interfaces fe-0/0/0 unit 0 family ethernet-switching port-mode trunk
vlan members all

[edit]
cjones@SW1# commit and-quit
commit complete
Exiting configuration mode
```

2. トランク設定を確認します。

```
cjones@SW1> show configuration interfaces fe-0/0/0.0
family ethernet-switching {
  port-mode trunk;
  vlan {
    members all;
  }
}
```

トランクインタフェースを確認するには

1. VLAN リストを表示します。

```
cjones@SW1> show vlans JUNOS_VLAN
Name      Tag      Interfaces
JUNOS_VLAN  10      fe-0/0/0.0
```

2. VLAN リストの詳細出力を表示して、VLAN が fe-0/0/0.0 でタグ付きの状態であることを確認します。

```
cjones@SW1> show vlans extensive JUNOS_VLAN
VLAN:JUNOS_VLAN, Created at:Sun Jun 24 12:54:01 2012
802.1Q Tag:10, Internal index:2, Admin State:Enabled, Origin:Static
Protocol:Port Mode, Mac aging time:300 seconds
Number of interfaces:Tagged 1 (Active = 0), Untagged 0 (Active = 0)
fe-0/0/0.0, tagged, trunk
```



### 3. イーサネットスイッチングインタフェースの詳細を表示します。

```
cjones@SW1> show ethernet-switching interfaces detail
Interface: fe-0/0/6.0, Index:84, State: up, Port mode:Trunk
Ether type for the interface:0x8100
VLAN membership:
    JUNOS_VLAN, 802.1Q Tag:10, tagged, msti-id:0, blocked by STP
```

### レイヤー3 インタフェース (Junos の RVI) を作成するには

RVI (Routed Virtual Interface) の設定は、IOS エンジニアには見慣れないものであるかもしれません。ただし、実際には、RVI の設定により、Junos インタフェースの設定で一貫性のあるアプローチが可能になっています。Junos では、ユニット番号が設定されている単一の VLAN インタフェースを使用します。ユニット番号は任意の値ですが、一貫性を確保するため、VLAN ID と一致させておくことが一般的な規則です。

このアプローチと、VLAN ごとに個別のインタフェースを作成する IOS のアプローチを比較してみてください。

#### 1. VLAN に L3 インタフェースを設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set vlans JUNOS_VLAN 13-interface vlan.10
```

2. RVI (VLAN インタフェース) を設定します。一般的な規則に従って、VLAN のユニット番号は VLAN ID と一致させます。この規則は必須ではありませんが、ベストプラクティスであることは確実です。

```
[edit]
cjones@R1# set interfaces vlan unit 10 family inet address 10.10.10.1/24
```

#### 3. 設定をコミットします。

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

#### 4. 設定を確認します。

```
cjones@R1> show configuration vlans
JUNOS_VLAN {
    vlan-id 10;
    13-interface vlan.10;
}

cjones@R1> show configuration interfaces vlan
unit 10 {
    family inet {
        address 10.10.10.1/24;
```

```
    }
}
```

### レイヤー 3 インタフェースを確認するには

1. スイッチのインタフェース IP アドレスのリストを表示して確認します。

```
cjones@R1> show interfaces terse | match vlan
vlan                up    up
vlan.10             up    up   inet    10.10.10.1/24
```

### まとめ

VLAN の設定と、VLAN へのレイヤー 3 インタフェースの設定のわかりやすさについては、Junos と IOS でほとんど変わりはありません。ここで、IOS エンジニアにとっての重要なポイントは、IOS の設定では、VLAN ID を作成してから、その VLAN の名前を設定できるのに対して、Junos では、VLAN の名前を設定して、その VLAN に基づいて VLAN-ID タグを設定するということです。

## シンプルな NAT

このセクションでは、エンタープライズ環境で目にすることが多い、外部インタフェースアドレスに対応する送信元 NAT と内部ホストに対応する宛先 NAT という最も一般的な 2 つの形式で NAT を設定しましょう。今回の比較では、2 つの OS に相違はないので、IOS エンジニアが注意する必要があるのは、単純にプロセス関連の事項だけになります。「Junos は IOS よりも何となく複雑だ」という声を耳にしますが、本書のほとんどのページで説明しているように、実際には、そのようなことはありません。

このセクションでは、図 2.4 のトポロジー図に示すように、単一の AS 内で直接接続された 2 台のルーターを使用します。

このセクションで実行するタスクのリストは以下のとおりです。

- 送信元 NAT 10.42.0.0/24 で外部インタフェース (200.200.200.1) を使用する。
- 200.200.200.1 のポート 80 の宛先 NAT で Web サーバー (10.42.0.10 のポート 80 で動作) を使用する。

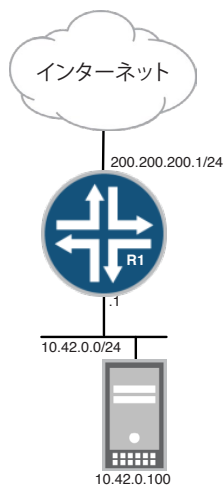


図 2.4 NAT トポロジー

## IOS の設定

*IOS で 10.42.0.0/24 の送信元 NAT を設定するには*

```
R1#configure terminal
R1(config)#interface f0/0
R1(config-if)#ip address 200.200.200.1 255.255.255.0
R1(config-if)#ip nat outside
R1(config-if)#interface f0/1
R1(config-if)#ip address 10.42.0.1 255.255.255.0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#access-list 1 permit 10.42.0.0 0.0.0.255
R1(config)#ip nat inside source list 1 interface FastEthernet 0/0 overload
R1(config)#end
R1#
```

*10.42.0.0/24 の送信元 NAT を確認するには*

```
R1# show ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
tcp 200.200.200.1:53638 10.42.0.2:53638  74.125.225.136:80 74.125.225.136:80
```

*IOS で Web サーバーの宛先 NAT を設定するには*

```
R1# configure terminal
R1(config)# ip nat inside source static tcp 10.42.0.100 80 200.200.200.1 80
extendable
R1(config)# end
R1#
```

### Web サーバーの宛先 NAT を確認するには

```
R1# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
tcp 200.200.200.1:80    10.42.0.100:80    ---                ---
```

## Junos の設定

今回は、同じ条件で Junos を設定します。ただし、本書で示すように NAT を設定するには、デバイスがフローモード（ブランチ向け SRX デバイスでは、このモードがデフォルト）であることが前提です。フローモードでは、トラフィックフローを許可するよう、ファイアウォールのセキュリティポリシーを設定する必要があります。セキュリティポリシーの設定については、本書の目的から外れるので説明しません。

さらに詳しくは セキュリティポリシーの設定の詳細については、ロブ・キャメロン、ブラッド・ウッドワード共著の『*Junos Security*』 (<http://www.juniper.net/books> で入手可能) を参照してください。

### Junos で 10.42.0.0/24 の送信元 NAT を設定するには

Junos の送信元 NAT はとても簡単です。標準的なポリシーの match/then 構文を使用し、NAT プロセスに関連するセキュリティゾーンを設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set interfaces fe-0/0/0 unit 0 family inet address 200.200.200.1/24

[edit]
cjones@R1# set interfaces fe-0/0/1 unit 0 family inet address 10.42.0.1/24

[edit]
cjones@R1# edit security nat source rule-set LAN_RULE

[edit security nat source rule-set LAN_RULE]
cjones@R1# set from zone LAN

[edit security nat source rule-set LAN_RULE]
cjones@R1# set to zone INET

[edit security nat source rule-set LAN_RULE]
cjones@R1# set rule NAT_10.42_24 match source-address 10.42.0.0/24

[edit security nat source rule-set LAN_RULE]
cjones@R1# set rule NAT_10.42_24 match destination-address 0.0.0.0/0

[edit security nat source rule-set LAN_RULE]
cjones@R1# set rule NAT_10.42_24 then source-nat interface
```

```
[edit security nat source rule-set LAN_RULE]
cjones@R1# top
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

以上の一連の set コマンドによって、以下の設定が作成されます。

```
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 200.200.200.1/24;
      }
    }
  }
  fe-0/0/1 {
    unit 0 {
      family inet {
        address 10.42.0.1/24;
      }
    }
  }
}
security {
  nat {
    source {
      rule-set LAN_RULE {
        from zone LAN;
        to zone INET;
        rule NAT_ALL {
          match {
            source-address 10.42.0.0/24;
            destination-address 0.0.0.0/0;
          }
          then {
            source-nat {
              interface;
            }
          }
        }
      }
    }
  }
}
```

## 10.42.0.0/24 の送信元 NAT を確認するには

1. 送信元 NAT のサマリーを表示します。

```
cjones@R1> show security nat source summary
Total port number usage for port translation pool:0
Maximum port number for port translation pool:16777216
Total pools:0

Total rules:1
Rule name      Rule set      From          To            Action
NAT_ALL        LAN_RULE      LAN           INET          interface
```

2. 10.42.0.100 のセキュリティフローセッションを表示します。

```
cjones@R1> show security flow session nat source-prefix 10.42.0.100
Session ID:14582, Policy name:PERMIT_ALL/4, Timeout:1754, Valid
  In:10.42.0.100/49401 -> 74.125.239.8/443;tcp, If: fe-0/0/1, Pkts:10, Bytes:394
  Out:74.125.239.8/443 -> 200.200.200.1/15654;tcp, If: fe-0/0/0.0, Pkts:8, Bytes:3970

Session ID:23401, Policy name:PERMIT_ALL/4, Timeout:1786, Valid
  In:10.42.0.100/49163 -> 207.17.137.239/80;tcp, If: fe-0/0/1, Pkts:18, Bytes:276
  Out:207.17.137.239/80 -> 200.200.200.1/5973;tcp, If: fe-0/0/0.0, Pkts:14, Bytes:353
Total sessions:2
```

## Junos で Web サーバーの宛先 NAT を設定するには

Junos では、宛先 NAT はシンプルかつ簡単です。Junos のセキュリティポリシーは標準的な if/then 形式に従っているので、設定の処理内容を正確に表現することが簡単です。

Junos の宛先 NAT には、宛先プールの作成が必要です。このプールによって、変換後、外部要求の送信先になる内部 IP アドレスを指定します。

宛先 NAT ポリシーでは、単純に一致条件と、一致条件を満たしたときの実行アクションを指定します。この例では、実行アクションに該当するのは、destination-nat pool <poolname> です。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set security nat destination pool WEBSERVER address 10.42.0.100/32

[edit]
cjones@R1# set security nat destination pool WEBSERVER address port 80

[edit]
cjones@R1# edit security nat destination rule-set DESTINATION_NAT

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# set from zone INET
```

```

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# set rule DST_NAT_WEBSERVER match source-address 0.0.0.0/0

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# set rule DST_NAT_WEBSERVER match destination-address 200.200.200.1/32

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# set rule DST_NAT_WEBSERVER match destination-port 80

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# set rule DST_NAT_WEBSERVER then destination-nat pool WEBSERVER

[edit security nat destination rule-set DESTINATION_NAT]
cjones@R1# top

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode

```

以上の一連の set コマンドによって、以下の設定が作成されます。

```

security {
  nat {
    destination {
      pool WEBSERVER {
        address 10.42.0.100/32 port 80;
      }
      rule-set DESTINATION_NAT {
        from zone INET;
        rule DST_NAT_WEBSERVER {
          match {
            source-address 0.0.0.0/0;
            destination-address 200.200.200.1/32;
            destination-port 80;
          }
          then {
            destination-nat pool WEBSERVER;
          }
        }
      }
    }
  }
}

```

## Web サーバーの宛先 NAT を確認するには

1. 宛先 NAT のサマリーを表示します。

```
cjones@R1> show security nat destination summary
Total pools:1
Pool name      Address          Routing          Port  Total
              Range           Instance        80    Address
WEBSERVER     10.42.0.100 - 10.42.0.100  default        80    1

Total rules:1
Rule name      Rule set         From              Action
DST_NAT_WEBSERVER  DESTINATION_NAT  INET              WEBSERVER
```

2. 10.42.0.100 への宛先 NAT フローのセキュリティセッションを表示します。

```
cjones@R1> show security flow session nat destination-prefix 200.200.200.1
Session ID:10057, Policy name:PERMIT_HTTP_TO_WEBSERVER/4, Timeout:1772, Valid
In:207.17.137.239/22072 --> 200.200.200.1/80;tcp, If: fe-0/0/0.0, Pkts:904,
Bytes:265591
Out:10.42.0.100/80 --> 207.17.137.239/22072;tcp, If: fe-0/0/0.1, Pkts:879,
Bytes:38638
```

## 結論

IP アドレスの変換前後で暗号のような名前を使用する IOS の場合よりも、Junos の NAT 設定は明らかに簡単です。ただし、NAT の確認は、わかりやすい NAT 変換テーブルのおかげで、IOS の方が簡単に行えます。



# 第3章

## ケーススタディ

<i>IOSの設定</i> .....	67
<i>Junosの設定</i> .....	76
<i>結論</i> .....	95

この章では、本書の設定例が実際の事例にどの程度、応用できるのかという点について学習します。前の各章のビルディングブロックを流用して、一般的なエンタープライズネットワークの規模を縮小したものを構築して設定します。

使用するのは、3台のルーターで構成されるシンプルなトポロジーであり、シングルエリア OSPF ネットワークとして設定され、EBGP で 2 系統のサービスプロバイダがそれぞれデフォルトルートを送信します。図 3.1 に、このトポロジー図を示します。

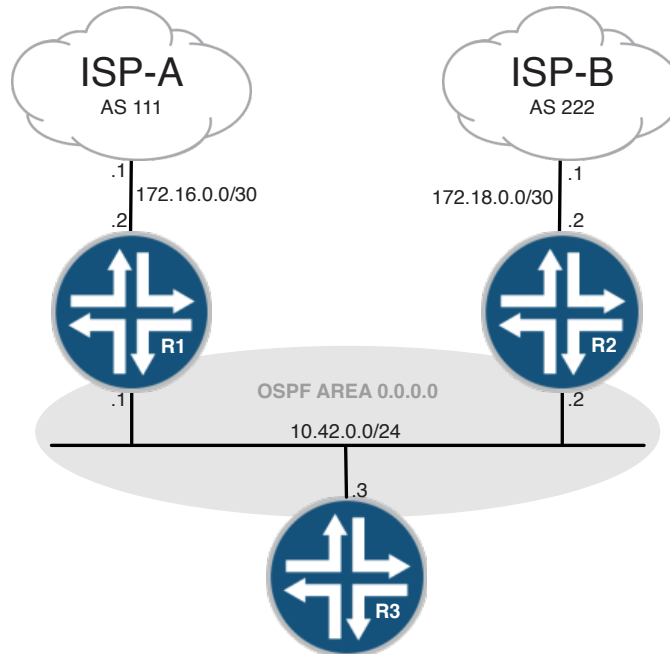


図 3.1 ケーススタディのトポロジー

このセクションで実行するタスクのリストは以下のとおりです。

- R1、R2、およびR3の間にOSPFを設定する。
- R1とR2のOSPFにデフォルトルートをアドバタイズする。
- R1とISP-Aの間にEBGPを設定する。
- R2とISP-Bの間にEBGPを設定する。
- next-hop-selfポリシーを含む、ループバックインタフェースを使用して、R1とR2の間にIBGPを設定する。
- ISP-Bを経由したインバウンドおよびアウトバウンドトラフィックを優先処理する。
- 10.42.0.0/16の集約プレフィックスをISP-AとISP-Bにアドバタイズする。

## IOS の設定

最初に、IOS を使用して、このネットワークを設定しましょう。これまでと同様に、重要なイベントは太字で表しています。

### IOS ルーターの初期接続を設定するには

1. R1 のインタフェースの IP アドレスを設定します。

```
R1# configure terminal
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/0
R1(config-if)# ip address 172.16.0.2 255.255.255.252
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/1
R1(config-if)# ip address 10.42.0.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

2. R2 のインタフェースの IP アドレスを設定します。

```
R2# configure terminal
R2(config)# interface loopback 0
R2(config-if)# ip address 2.2.2.2 255.255.255.255
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/0
R2(config-if)# ip address 172.18.0.2 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# interface FastEthernet 0/1
R2(config-if)# ip address 10.42.0.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# end
R2#
```

3. R3 のインタフェースの IP アドレスを設定します。

```
R3# configure terminal
R3(config)# interface loopback 0
R3(config-if)# ip address 3.3.3.3 255.255.255.255
R3(config-if)# no shutdown
R3(config-if)# interface FastEthernet 0/0
R3(config-if)# ip address 10.42.0.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# end
R3#
```

### 初期接続を確認するには

1. R1 から R2 への ping を実行します。

```
R1#ping 10.42.0.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.0.2, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R1#
```

2. R1 から R3 への ping を実行します。

```
R1#ping 10.42.0.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.0.3, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R1#
```

3. R2 から R3 への ping を実行します。

```
R2#ping 10.42.0.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.42.0.3, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R2#
```

4. R1 から ISP-A への ping を実行します。

```
R1#ping 172.16.0.1
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.16.0.1, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R1#
```

5. R2 から ISP-B への ping を実行します。

```
R2#ping 172.18.0.1
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.18.0.1, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms  
R2#
```

### *IOS ルーターを OSPF 用に設定するには*

1. OSPF プロセスを設定して、インタフェースを R3 のエリア 0 に割り当てます。ループバックインタフェースをパッシブに設定します。また、OSPF のルーター ID を手動で設定します。

```
R1# configure terminal
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 1.1.1.1 0.0.0.0 area 0
R1(config-router)# network 10.42.0.1 0.0.0.0 area 0
R1(config-router)# passive-interface lo0
R1(config-router)# end
R1#
```

2. OSPF プロセスを設定して、インタフェースを R3 のエリア 0 に割り当てます。また、OSPF のルーター ID を手動で設定します。

```
R2# configure terminal
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 2.2.2.2 0.0.0.0 area 0
R2(config-router)# network 10.42.0.2 0.0.0.0 area 0
R2(config-router)# passive-interface lo0
R2(config-router)# end
R2#
```

3. OSPF プロセスを設定して、インタフェースを R3 のエリア 0 に割り当てます。また、OSPF のルーター ID を手動で設定します。

```
R3# configure terminal
R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 3.3.3.3 0.0.0.0 area 0
R3(config-router)# network 10.42.0.3 0.0.0.0 area 0
R3(config-router)# passive-interface lo0
R3(config-router)# end
R3#
```

### OSPF の設定を確認するには

1. R1 の OSPF 隣接関係を確認します。

```
R1#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/BDR	00:00:32	10.42.0.2	FastEthernet0/1
3.3.3.3	1	FULL/DR	00:00:34	10.42.0.3	FastEthernet0/1

```
R1#
```

2. R2 の OSPF 隣接関係を確認します。

```
R2#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/DROTHER	00:00:30	10.42.0.1	FastEthernet0/1
3.3.3.3	1	FULL/BDR	00:00:33	10.42.0.3	FastEthernet0/1

```
R2#
```

3. R3 の OSPF 隣接関係を確認します。

```
R3#show ip ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/DROTHER	00:00:37	10.42.0.1	FastEthernet0/0
2.2.2.2	1	FULL/BDR	00:00:35	10.42.0.2	FastEthernet0/0

```
R3#
```

*IOS でデフォルトルートを OSPF に注入するよう R1 と R2 を設定するには*

1. デフォルトルートを OSPF プロセスに注入するよう R1 を設定します。

```
R1#configure terminal
R1(config)#router ospf 1
R1(config-router)#default-information originate
R1(config-router)#end
R1#
```

2. デフォルトルートを OSPF プロセスに注入するよう R2 を設定します。

```
R2#configure terminal
R2(config)#router ospf 1
R2(config-router)#default-information originate
R2(config-router)#end
R2#
```

*IOS で OSPF のデフォルトルートを確認するには*

1. R3 の OSPF データベースで OSPF のタイプ 5 LSA をチェックします。

```
R3#show ip ospf database external
```

```
OSPF Router with ID (3.3.3.3) (Process ID 1)
```

```
Type-5 AS External Link States
```

```
Routing Bit Set on this LSA
LS age:164
Options:(No TOS-capability, DC)
LS Type:AS External Link
Link State ID:0.0.0.0 (External Network Number )
Advertising Router:1.1.1.1
LS Seq Number:80000001
Checksum:0x1D91
Length:36
Network Mask:/0
Metric Type:2 (Larger than any link state path)
TOS:0
Metric:1
Forward Address:0.0.0.0
External Route Tag:1
```

```
Routing Bit Set on this LSA
LS age:82
Options:(No TOS-capability, DC)
```

```

LS Type:AS External Link
Link State ID:0.0.0.0 (External Network Number )
Advertising Router:2.2.2.2
LS Seq Number:80000001
Checksum:0xFEAB
Length:36
Network Mask:/0
    Metric Type:2 (Larger than any link state path)
    TOS:0
    Metric:1
    Forward Address:0.0.0.0
    External Route Tag:1

```

2. RIB で OSPF の外部ルートをチェックします。

```

R3#show ip route ospf
    1.0.0.0/32 is subnetted, 1 subnets
0       1.1.1.1 [110/11] via 10.42.0.1, 00:59:55, FastEthernet0/0
    2.0.0.0/32 is subnetted, 1 subnets
0       2.2.2.2 [110/11] via 10.42.0.2, 00:59:55, FastEthernet0/0
O*E2 0.0.0.0/0 [110/1] via 10.42.0.2, 00:03:00, FastEthernet0/0
        [110/1] via 10.42.0.1, 00:04:22, FastEthernet0/0

```

*IOS ルーター R1 と R2 を EBGP で ISP-A と ISP-B に設定するには*

1. R1 (AS 65001) を ISP-A (AS 111) のピアに設定します。

```

R1# configure terminal
R1(config)# router bgp 65001
R1(config-router)# no synchronization
R1(config-router)# neighbor 172.16.0.1 remote-as 111
R1(config-router)# no auto-summary
R1(config-router)# end
R1#

```

2. R2 (AS 65001) を ISP-B (AS 222) のピアに設定します。

```

R2# configure terminal
R2(config)# router bgp 65001
R2(config-router)# no synchronization
R2(config-router)# neighbor 172.18.0.1 remote-as 222
R2(config-router)# no auto-summary
R2(config-router)# end
R2#

```

*IOS の EBGP 設定を確認するには*

1. R1 の EBGP で ISP-A とのピアリングを確認します。

```

R1#show ip bgp neighbors 172.16.0.1 | include =
    BGP state = Established, up for 00:08:37

R1#show ip bgp summary | include 172.16.0.1|Neighbor
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
172.16.0.1    4  111     11     10      2    0    0 00:07:50      1
R1#

```

2. R1 の Adj-RIB-In テーブル (BGP テーブル) を調べて、ISP-A によってアドバタイズされたデフォルトルートを R1 が受信していることを確認します。

```
R1#show ip bgp
BGP table version is 2, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ?- incomplete
```

```
   Network          Next Hop          Metric LocPrf Weight Path
*> 0.0.0.0          172.16.0.1        0             0 111 i
```

3. BGP で学習されたデフォルトルートが RIB (ルーティングテーブル) にインストールされていることを確認します。

```
R1#show ip route bgp
B* 0.0.0.0/0 [20/0] via 172.16.0.1, 00:23:52
```

4. R2 の EBGP で ISP-B とのピアリングを確認します。

```
R2#show ip bgp neighbors 172.18.0.1 | include =
BGP state = Established, up for 00:02:46
```

```
R2#show ip bgp summary | include 172.18.0.1|Neighbor
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
172.18.0.1    4  222     7      6       2    0    0 00:03:37      1
```

5. R2 の Adj-RIB-In テーブルを調べて、ISP-B によってアドバタイズされたデフォルトルートを R2 が受信していることを確認します。

```
R2#show ip bgp
BGP table version is 2, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ?- incomplete
```

```
   Network          Next Hop          Metric LocPrf Weight Path
*> 0.0.0.0          172.18.0.1        0             0 222 i
```

6. BGP で学習されたデフォルトルートが RIB (ルーティングテーブル) にインストールされていることを確認します。

```
R2#show ip route bgp
B* 0.0.0.0/0 [20/0] via 172.18.0.1, 00:21:23
```

### IOS ルーター R1 と R2 を IBGP 用に設定するには

1. ピアリング用のループバックインタフェースを使用して、IBGP の R1 を R2 に設定します。



```
R1# configure terminal
R1(config)# router bgp 65001
R1(config-router)# neighbor 2.2.2.2 remote-as 65001
R1(config-router)# neighbor 2.2.2.2 update-source Loopback0
R1(config-router)# end
R1#
```

2. R2 のループバックへの完全な到達性を確保するため、R1 のスタティックルートを設定します。OSPF が動作しているため、このステップは必須ではありませんが、デモ目的で設定することにします。OSPF ルートの使用が優先されるため、スタティックルートに高いルート優先順位 (IOS の *Administrative Distance*) を設定します。

```
R1# configure terminal
R1(config)#ip route 2.2.2.2 255.255.255.255 10.42.0.2 254
R1(config)#end
R1#
```

3. R2 に送信されたすべてのプレフィックスのネクストホップを更新するよう、R1 の IBGP ピアリングを設定します。

```
R1#configure terminal
R1(config)#router bgp 65001
R1(config-router)#neighbor 2.2.2.2 next-hop-self
R1(config-router)#end
R1#
```

4. ピアリング用のループバックインタフェースを使用して、IBGP の R2 を R1 に設定します。

```
R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)#neighbor 1.1.1.1 remote-as 65001
R2(config-router)#neighbor 1.1.1.1 update-source Loopback0
R2(config)#end
R2#
```

5. 万一の OSPF の障害発生時に、R1 の 5 で設定したループバックへの完全な到達性を確保するため、R2 のスタティックルートを設定します。

```
R2#configure terminal
R2(config)#ip route 1.1.1.1 255.255.255.255 10.42.0.1 254
R2(config)#end
R2#
```

6. R1 に送信されたすべてのプレフィックスのネクストホップを更新するよう、R2 の IBGP ピアリングを設定します。

```
R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)#neighbor 1.1.1.1 next-hop-self
```

```
R2(config-router)#end
R2#
```

### R1とR2の間のBGP設定を確認するには

1. R1とR2の間のIBGPの隣接関係を確認します。

```
R1#show ip bgp neighbors 2.2.2.2 | include =
  BGP state = Established, up for 00:09:14
```

2. R1がR2からBGPプレフィックスを受信していることを確認します。

```
R1#show ip bgp summary | include 2.2.2.2|Neighbor
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
2.2.2.2       4 65001    12     11      2    0    0 00:06:30      1
```

### 集約プレフィックス10.42.0.0/16をAS111とAS222にアドバタイズするには

1. aggregate-addressコマンドをR1のBGPプロセスに追加します。

```
R1#configure terminal
R1(config)#router bgp 65001
R1(config-router)# aggregate-address 10.42.0.0 255.255.0.0 summary-only
R1(config-router)# network 10.42.0.0 mask 255.255.255.0
R1(config-router)# end
R1#
```

2. aggregate-addressコマンドをR2のBGPプロセスに追加します。

```
R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)# aggregate-address 10.42.0.0 255.255.0.0 summary-only
R2(config-router)# network 10.42.0.0 mask 255.255.255.0
R2(config-router)# end
R2#
```

### 集約ルートがAS111とAS222に送信されたことを確認するには

1. R1からISP-Aにアドバタイズされているルートのリストをチェックします。

```
R1#show ip bgp neighbors 172.16.0.1 advertised-routes
```

```
BGP table version is 5, local router ID is 1.1.1.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```



### アウトバウンドトラフィックを優先してISP-B経由でネットワークから送信するには

1. ISP-Bから学習された全ルートのLocal Preferenceを上げるよう、R2を設定します。

```
R2#configure terminal
R2(config)#route-map PREFER_ISPB_OUTBOUND permit 10
R2(config-route-map)#set local-preference 110
R2(config-route-map)#router bgp 65001
R2(config-router)#neighbor 172.18.0.1 route-map PREFER_ISPB_OUTBOUND in
R2(config-router)#end
R2#
```

### Local Preferenceの値が変更されたことを確認するには

```
R2#show ip bgp | include Network|172.18.0.1
Network          Next Hop          Metric LocPrf Weight Path
* 0.0.0.0         172.18.0.1       0      110     0 222 i
```

## Junos の設定

今度は、同じ一連の条件をJunosで設定しましょう。IOSエンジニアは、JunosではIGPとBGP両方のルーティング操作のほとんどをポリシー設定に依存している点に注目してください。

このセクションの作業が終了したら、最終的な設定を確認します。実際に、設定が論理的で整理されていることに気が付くことでしょう。さらに、Junosの設定は設定の目的を問わず、かなりの部分が標準形式に従っているので、予測しやすいという特徴があります。

### Junos ルーターの初期接続を設定するには

1. R1のインタフェースのIPアドレスを設定します。

```
cjones@R1> configure
[edit]
cjones@R1# set interfaces ge-0/0/0 unit 0 family inet address 172.16.0.2/30

[edit]
cjones@R1# set interfaces ge-0/0/1 unit 0 family inet address 10.42.0.1/24

[edit]
cjones@R1# set interfaces lo0 unit 0 family inet address 1.1.1.1/32

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
cjones@R1>
```

2. R1のIPアドレスを確認します。

```
cjones@R1> show configuration interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
```

```
        address 172.6.0.2/30;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.42.0.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 1.1.1.1/24;
        }
    }
}
}
```

3. R2 のインタフェースの IP アドレスを設定します。

```
cjones@R2> configure
[edit]
cjones@R2# set interfaces ge-0/0/0 unit 0 family inet address 172.18.0.2/30

[edit]
cjones@R2# set interfaces ge-0/0/1 unit 0 family inet address 10.42.0.2/24

[edit]
cjones@R2# set interfaces lo0 unit 0 family inet address 2.2.2.2/32

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
cjones@R2>
```

4. R2 の IP アドレスを確認します。

```
cjones@R2> show configuration interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 172.18.0.2/30;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.42.0.2/24;
        }
    }
}
lo0 {
    unit 0 {
```

```
        family inet {  
            address 2.2.2.2/24;  
        }  
    }  
}
```

5. R3 のインタフェースの IP アドレスを設定します。

```
cjones@R3> configure  
[edit]  
cjones@R3# set interfaces ge-0/0/1 unit 0 family inet address 10.42.0.3/24  
  
[edit]  
cjones@R3# set interfaces lo0 unit 0 family inet address 3.3.3.3/32  
  
[edit]  
cjones@R2# commit and-quit  
commit complete  
Exiting configuration mode  
cjones@R3>
```

6. R3 の IP アドレスを確認します。

```
cjones@R3> show configuration interfaces  
ge-0/0/1 {  
    unit 0 {  
        family inet {  
            address 10.42.0.3/24;  
        }  
    }  
}  
lo0 {  
    unit 0 {  
        family inet {  
            address 3.3.3.3/32;  
        }  
    }  
}
```

*初期接続を確認するには*

1. R1 から R2 への ping を実行します。

```
cjones@R1> ping 10.42.0.2 rapid  
PING 10.42.0.2 (10.42.0.2):56 data bytes  
!!!!  
--- 10.42.0.2 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 2.238/2.791/4.261/0.752 ms
```

2. R1 から R3 への ping を実行します。

```
cjones@R1> ping 10.42.0.3 rapid  
PING 10.42.0.3 (10.42.0.3):56 data bytes
```

```
!!!!  
--- 10.42.0.3 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 2.332/5.504/17.629/6.063 ms
```

3. R2 から R3 への ping を実行します。

```
cjones@R2> ping 10.42.0.3 rapid  
PING 10.42.0.3 (10.42.0.3):56 data bytes  
!!!!  
--- 10.42.0.3 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 2.042/7.932/30.717/11.394 ms
```

4. R1 から ISP-A への ping を実行します。

```
cjones@R1> ping 172.16.0.1 rapid  
PING 172.16.0.1 (172.16.0.1):56 data bytes  
!!!!  
--- 172.16.0.1 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 2.625/7.827/31.212/10.268 ms
```

5. R2 から ISP-B への ping を実行します。

```
cjones@R2> ping 172.18.0.1 rapid  
PING 172.18.0.1 (172.18.0.1):56 data bytes  
!!!!  
--- 172.18.0.1 ping statistics ---  
5 packets transmitted, 5 packets received, 0% packet loss  
round-trip min/avg/max/stddev = 2.766/6.983/29.472/11.121 ms
```

### *Junos ルーターを OSPF 用に設定するには*

1. 今度は、R1 の OSPF エリア 0 にインタフェースを設定しましょう。ループバックインタフェースをパッシブに設定して、OSPF のルーター ID を手動で設定します。

```
cjones@R1> configure  
[edit]  
cjones@R1# set protocols ospf area 0 interface ge-0/0/1.0  
  
[edit]  
cjones@R1# set protocols ospf area 0 interface lo0.0 passive  
  
[edit]  
cjones@R1# set routing-options router-id 1.1.1.1  
  
[edit]  
cjones@R1# commit and-quit  
commit complete  
Exiting configuration mode  
cjones@R1>
```

2. R1 の OSPF 設定を確認します。

```
cjones@R1> show configuration routing-options
router-id 1.1.1.1;
```

```
cjones@R1> show configuration protocols
ospf {
  area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface lo0.0 {
      passive;
    }
  }
}
```

3. 今度は、R2 の OSPF エリア 0 の R2 インタフェースで同じ手順を実行しましょう。ループバックインタフェースをパッシブに設定して、OSPF のルーター ID を手動で設定します。

```
cjones@R2> configure
[edit]
cjones@R2# set protocols ospf area 0 interface ge-0/0/1.0

[edit]
cjones@R2# set protocols ospf area 0 interface lo0.0 passive

[edit]
cjones@R2# set routing-options router-id 2.2.2.2

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
cjones@R2>
```

4. R2 の OSPF 設定を確認します。

```
cjones@R2> show configuration routing-options
router-id 2.2.2.2;
```

```
cjones@R2> show configuration protocols
ospf {
  area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface lo0.0 {
      passive;
    }
  }
}
```

5. R3 についても同じ手順を実行します。R3 の OSPF エリア 0 にインタフェースを設定します。ループバックインタフェースをパッシブに設定して、OSPF のルーター ID を手動で設定します。

```
cjones@R3> configure
[edit]
cjones@R3# set protocols ospf area 0 interface ge-0/0/1.0
```



```
[edit]
cjones@R3# set protocols ospf area 0 interface lo0.0 passive

[edit]
cjones@R3# set routing-options router-id 3.3.3.3

[edit]
cjones@R3# commit and-quit
commit complete
Exiting configuration mode
cjones@R3>
```

6. R3 の OSPF 設定を確認します。

```
cjones@R3> show configuration routing-options
router-id 3.3.3.3;

cjones@R3> show configuration protocols
ospf {
  area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface lo0.0 {
      passive;
    }
  }
}
```

### OSPF の設定を確認するには

1. R1 の OSPF 隣接関係を確認します。

```
cjones@R1> show ospf neighbor
Address      Interface    State      ID          Pri  Dead
10.42.0.3    ge-0/0/1.0  Full      3.3.3.3    128  30
10.42.0.2    ge-0/0/1.0  Full      2.2.2.2    128  37
```

2. R2 の OSPF 隣接関係を確認します。

```
cjones@R2> show ospf neighbor
Address      Interface    State      ID          Pri  Dead
10.42.0.1    ge-0/0/1.0  Full      1.1.1.1    128  32
10.42.0.3    ge-0/0/1.0  Full      3.3.3.3    128  30
```

3. R3 の OSPF 隣接関係を確認します。

```
cjones@R3> show ospf neighbor
Address      Interface    State      ID          Pri  Dead
10.42.0.1    ge-0/0/1.0  Full      1.1.1.1    128  32
10.42.0.2    ge-0/0/1.0  Full      2.2.2.2    128  37
```

*Junos* でデフォルトルートを OSPF に注入するよう R1 と R2 を設定するには

1. R1 のスタティックデフォルトルートを設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options static route 0/0 discard
```

2. R1 のエクスポートポリシーを設定します。

```
[edit]
cjones@R1# edit policy-options policy-statement DEFAULT_TO_OSPF

[edit policy-options policy-statement DEFAULT_TO_OSPF]
cjones@R1# set from protocol static

[edit policy-options policy-statement DEFAULT_TO_OSPF]
cjones@R1# set from route-filter 0/0 exact

[edit policy-options policy-statement DEFAULT_TO_OSPF]
cjones@R1# set then accept

[edit policy-options policy-statement DEFAULT_TO_OSPF]
cjones@R1# top
```

```
[edit]
cjones@R1# show policy-options
policy-statement DEFAULT_TO_OSPF {
  from {
    protocol static;
    route-filter 0.0.0.0/0 exact;
  }
  then accept;
}
```

3. デフォルトルートを OSPF プロセスに注入するよう R1 のエクスポートポリシーを設定します。

```
[edit]
cjones@R1# set protocols ospf export DEFAULT_TO_OSPF

[edit]
cjones@R1# show protocols ospf
export DEFAULT_TO_OSPF;
area 0.0.0.0 {
  interface ge-0/0/1.0;
  interface lo0.0 {
    passive;
  }
}
```

4. R1 の設定を確認してコミットします。

```
[edit]
cjones@R1# show | compare
```

```

[edit routing-options]
+ static {
+   route 0.0.0.0/0 discard;
+ }
[edit protocols ospf]
+ export DEFAULT_TO_OSPF;
[edit]
+ policy-options {
+   policy-statement DEFAULT_TO_OSPF {
+     from {
+       protocol static;
+       route-filter 0.0.0.0/0 exact;
+     }
+     then accept;
+   }
+ }

[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode

```

5. 次に、同じ設定を R2 に追加する必要があります。

ここでは、新しい設定を試しましょう。同じ内容を入力し直す代わりに、ステップ 4 の patch 構文を使用して R2 を設定します。上の例から compare の出力をコピーして、R2 で load patch terminal コマンドを使用します。貼り付けて、Ctrl+D キーを押します。次に、以下のよう  
に、変更内容をコミットします。

```

cjones@R2> configure
Entering configuration mode

[edit]
cjones@R2# load patch terminal
[Type ^D at a new line to end input]
[edit routing-options]
+ static {
+   route 0.0.0.0/0 discard;
+ }
[edit protocols ospf]
+ export DEFAULT_TO_OSPF;
[edit]
+ policy-options {
+   policy-statement DEFAULT_TO_OSPF {
+     from {
+       protocol static;
+       route-filter 0.0.0.0/0 exact;
+     }
+     then accept;
+   }
+ }|
load complete

```

```
[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

### Junos で OSPF のデフォルトルートを確認するには

1. R3 の OSPF データベースで OSPF のタイプ 5 LSA をチェックします。

```
cjones@R3> show ospf database external
OSPF AS SCOPE link state database
Type      ID          Adv Rtr      Seq         Age  Opt  Cksum  Len
Extern    0.0.0.0     1.1.1.1     0x80000001  599  0x22 0xe2cb 36
Extern    0.0.0.0     2.2.2.2     0x80000001  220  0x22 0xc4e5 36
```

R3 の RIB で OSPF の外部ルートをチェックします。

注 この例では、exact キーワードを使用しています。このキーワードを使用しなければ、全ルートが 0.0.0.0/0 と一致するので、RIB 全体が表示されます。

```
cjones@R3> show route 0.0.0.0/0 exact

inet.0:7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[OSPF/150] 00:04:35, metric 0, tag 0
                   to 10.42.0.1 via ge-0/0/1.0
                   > to 10.42.0.2 via ge-0/0/1.0
```

### Junos ルーター R1 と R2 を EBGP で ISP-A と ISP-B に設定するには

1. R1 (AS 65001) を ISP-A (AS 111) のピアに設定します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options autonomous-system 65001

[edit]
cjones@R1# set protocols bgp group ISP-A type external neighbor 172.16.0.1 peer-as 111
```

2. R1 の BGP 設定を確認します。

```
[edit]
cjones@R1# show routing-options autonomous-system
65001;

[edit]
cjones@R1# show protocols bgp
group ISP-A {
    type external;
    neighbor 172.16.0.1 {
        peer-as 111;
    }
}
```

3. R1 の BGP 設定をコミットします。

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

4. R2 (AS 65001) を ISP-B (AS 222) のピアに設定します。

```
cjones@R2> configure
Entering configuration mode
```

```
[edit]
cjones@R2# set routing-options autonomous-system 65001
```

```
[edit]
cjones@R2# set protocols bgp group ISP-B type external neighbor 172.18.0.1 peer-as 222
```

5. R2 の BGP 設定を確認します。

```
[edit]
cjones@R2# show routing-options autonomous-system 65001;
```

```
[edit]
cjones@R2# show protocols bgp
group ISP-B {
    type external;
    neighbor 172.18.0.1 {
        peer-as 222;
    }
}
```

6. R2 の BGP 設定をコミットします。

```
[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

### IOS の EBGW 設定を確認するには

1. R1 の EBGW で ISP-A とのピアリングを確認します。

```
cjones@R1> show bgp neighbor 172.16.0.1
Peer:172.16.0.1+57730 AS 111 Local:172.16.0.2+179 AS 65001
Type:External State:Established Flags:<ImportEval Sync>
Last State:OpenConfirm Last Event:RecvKeepAlive
Last Error:None
Options:<Preference PeerAS Refresh>
Holdtime:90 Preference:170
Number of flaps:0
Peer ID:172.16.0.1 Local ID:1.1.1.1 Active Holdtime:90
Keepalive Interval:30 Peer index:0
BFD: disabled, down
Local Interface:ge-0/0/0.0
```

2. R1 の Adj-RIB-In テーブルを調べて、ISP-A によってアドバタイズされた 111.111.111.0/24 ルートを R1 が受信していることを確認します。

```
cjones@R1> show route receive-protocol bgp 172.16.0.1
```

```
inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 111.111.111.0/24      172.16.0.1                                111 I
```

3. ISP-A の BGP で学習されたルートが R1 の RIB にインストールされていることを確認します。

```
cjones@R1> show route protocol bgp
```

```
inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

111.111.111.0/24  *[BGP/170] 00:01:30, localpref 100
                  AS path:111 I
                  > to 172.16.0.1 via ge-0/0/0.0
```

4. R2 の EBGP で ISP-B とのピアリングを確認します。

```
cjones@R2> show bgp neighbor 172.18.0.1
```

```
Peer:172.18.0.1+179 AS 222 Local:172.18.0.2+56620 AS 65001
  Type:External State:Established Flags:<ImportEval Sync>
  Last State:OpenConfirm Last Event:RecvKeepAlive
  Last Error:None
  Options:<Preference PeerAS Refresh>
  Holdtime:90 Preference:170
  Number of flaps:0
  Peer ID:172.18.0.1 Local ID:2.2.2.2 Active Holdtime:90
  Keepalive Interval:30 Peer index:0
  BFD: disabled, down
  Local Interface:ge-0/0/0.0
```

5. R2 の Adj-RIB-In テーブルを調べて、ISP-B によってアドバタイズされた 111.111.111.0/24 ルートを R2 が受信していることを確認します。

```
cjones@R2> show route receive-protocol bgp 172.18.0.1
```

```
inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 111.111.111.0/24      172.18.0.1                                222 I
```

6. ISP-B の BGP で学習されたルートが R2 の RIB にインストールされていることを確認します。

```
cjones@R2> show route protocol bgp
```

```
inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

111.111.111.0/24  *[BGP/170] 00:02:42, localpref 100
                  AS path:222 I
                  > to 172.18.0.1 via ge-0/0/0.0
```

### Junos ルーター R1 と R2 を IBGP 用に設定するには

1. ピアリング用のループバックインタフェースを使用して、IBGP の R1 を R2 に設定します。

```
cjones@R1> configure
Entering configuration mode
```

```
[edit]
cjones@R1# set protocols bgp group IBGP type internal neighbor 2.2.2.2 local-address
1.1.1.1 peer-as 65001
```

2. R2 のループバックへの完全な到達性を確保するため、R1 のスタティックルートを設定します。

OSPF が動作しているので、このステップは必須ではありませんが、デモ目的で設定することにします。OSPF ルートの使用が優先されるので、スタティックルートに高いルート優先順位を設定します。フローティングスタティックルートが使用されるのは、IGP の障害が発生した場合に限定されます。

```
[edit]
cjones@R1# set routing-options static route 2.2.2.2/32 next-hop 10.42.0.2 preference
254
```

3. R2 に送信されたすべての IBGP プレフィックスのネクストホップを専用のインタフェースアドレスに変更するよう、R1 のポリシーを設定します。

```
[edit]
cjones@R1# set policy-options policy-statement NHS then next-hop self
```

4. R2 への IBGP ピアリングで *NHS* ポリシーを使用するよう、R1 を設定します。

```
[edit]
cjones@R1# set protocols bgp group IBGP export NHS
```

5. R1 の変更内容を確認して、設定をコミットします。

```
[edit]
cjones@R1# show | compare
[edit routing-options static]
    route 0.0.0.0/0 { ...}
+   route 2.2.2.2/32 {
+       next-hop 10.42.0.2;
+       preference 254;
+   }
[edit protocols bgp]
    group ISP-A { ...}
+   group IBGP {
+       type internal;
+       export NHS;
+       neighbor 2.2.2.2 {
+           local-address 1.1.1.1;
+           peer-as 65001;
+       }
+   }
```

```
[edit policy-options]
+ policy-statement NHS {
+   then {
+     next-hop self;
+   }
+ }
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

6. ピアリング用のループバックインタフェースを使用して、IBGP の R2 を R1 に設定します。

```
cjones@R2> configure
Entering configuration mode
```

```
[edit]
cjones@R2# set protocols bgp group IBGP type internal neighbor 1.1.1.1 local-address 2.2.2.2 peer-as 65001
```

7. R1 のループバックへの完全な到達性を確保するため、R2 のスタティックルートを設定します。

```
[edit]
cjones@R2# set routing-options static route 1.1.1.1/32 next-hop 10.42.0.1 preference 254
```

8. R1 に送信されたすべての IBGP プレフィックスのネクストホップを専用のインタフェースアドレスに変更するよう、R2 のポリシーを設定します。

```
[edit]
cjones@R2# set policy-options policy-statement NHS then next-hop self
```

9. R1 への IBGP ピアリングで *NHS* ポリシーを使用するよう、R2 を設定します。

```
[edit]
cjones@R2# set protocols bgp group IBGP export NHS
```

10. R1 の変更内容を確認して、設定をコミットします。

```
[edit]
cjones@R2# show | compare
[edit routing-options static]
  route 0.0.0.0/0 { ...}
+ route 1.1.1.1/32 {
+   next-hop 10.42.0.1;
+   preference 254;
+ }
[edit protocols bgp]
  group ISP-B { ...}
+ group IBGP {
+   type internal;
+   export NHS;
```



```

+     neighbor 1.1.1.1 {
+         local-address 2.2.2.2;
+         peer-as 65001;
+     }
+ }
[edit policy-options]
+ policy-statement NHS {
+     then {
+         next-hop self;
+     }
+ }

[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode

```

### R1 と R2 の間の BGP 設定を確認するには

1. R1 と R2 の間の IBGP の隣接関係を確認します。

```

cjones@R1> show bgp neighbor 2.2.2.2
Peer:2.2.2.2+63702 AS 65001 Local:1.1.1.1+179 AS 65001
Type:Internal State:Established Flags:<Sync>
Last State:OpenConfirm Last Event:RcvKeepAlive
Last Error:None
Export:[ NHS ]
Options:<Preference LocalAddress PeerAS Refresh>
Local Address:1.1.1.1 Holdtime:90 Preference:170
Number of flaps:0
Peer ID:2.2.2.2 Local ID:1.1.1.1 Active Holdtime:90

```

2. R1 が正しいネクストホップで R2 からプレフィックスを受信していることを確認します。

```

cjones@R1> show route receive-protocol bgp 2.2.2.2

inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref      AS path
* 111.111.111.0/24      2.2.2.2      100      222 I

```

3. R2 からのプレフィックスが R1 の RIB にインストールされていることを確認します。

```

cjones@R1> show route protocol bgp 111.111.111.0/24

inet.0:10 destinations, 13 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

111.111.111.0/24  *[BGP/170] 00:21:51, localpref 100
                  AS path:111 I
                  > to 172.16.0.1 via ge-0/0/0.0
                  [BGP/170] 00:02:18, localpref 100, from 2.2.2.2
                  AS path:222 I
                  > to 10.42.0.2 via ge-0/0/1.0

```

4. R2 が正しいネクストホップで R1 からプレフィックスを受信していることを確認します。

```
cjones@R2> show route receive-protocol bgp 1.1.1.1

inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 111.111.111.0/24      1.1.1.1              100      111 I
```

5. R1 からのプレフィックスが R2 の RIB にインストールされていることを確認します。

```
cjones@R2> show route protocol bgp 111.111.111.0/24

inet.0:10 destinations, 13 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

111.111.111.0/24  *[BGP/170] 00:03:28, localpref 100
                  AS path:222 I
                  > to 172.18.0.1 via ge-0/0/0.0
                  [BGP/170] 00:23:01, localpref 100, from 1.1.1.1
                  AS path:111 I
                  > to 10.42.0.1 via ge-0/0/1.0
```

### 集約プレフィックス 10.42.0.0/16 を AS 111 と AS 222 にアドバタイズするには

Junos では、集約ルートのアドバタイズはかなりシンプルです。必要な設定は IOS の場合よりも多いですが、これまで見てきたように同じポリシー構造に従っています。

集約ルートをアドバタイズするには、その集約ルートを先に作成しておく必要があります。具体的には、スタティックルートの場合と同じ構文を使用します。作成した集約ルートはポリシーで照合して、そのポリシーを BGP のエクスポートポリシーとして適用します。

重要なポイントとして、BGP が集約ルート以外の情報をアドバタイズしないよう、ポリシーの末尾に reject ステートメントが必要です。

1. R1 の集約ルートを作成します。

```
cjones@R1> configure
Entering configuration mode

[edit]
cjones@R1# set routing-options aggregate route 10.42.0.0/16
```

2. 照合で一致した集約ルートを受け入れて、それ以外のルートは拒否するよう、R1 のポリシーを設定します。

```
[edit]
cjones@R1# edit policy-options policy-statement AGG_TO_ISP

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R1# set term ACCEPT_AGG from protocol aggregate
```

```
[edit policy-options policy-statement AGG_TO_ISP]
cjones@R1# set term ACCEPT_AGG from route-filter 10.42.0.0/16 exact
```

```
[edit policy-options policy-statement AGG_TO_ISP]
cjones@R1# set term ACCEPT_AGG then accept
```

```
[edit policy-options policy-statement AGG_TO_ISP]
cjones@R1# set term REJECT_OTHERS then reject
```

```
[edit policy-options policy-statement AGG_TO_ISP]
cjones@R1# top
```

3. BGP 設定のグループまたは隣接機器の export ステートメントにポリシーを設定します。

```
[edit]
cjones@R1# set protocols bgp group ISP-A neighbor 172.16.0.1 export AGG_TO_ISP
```

4. R1 の変更内容を確認してコミットします。

```
[edit]
cjones@R1# show | compare
[edit routing-options]
+ aggregate {
+   route 10.42.0.0/16;
+ }
[edit protocols bgp group ISP-A neighbor 172.16.0.1]
+ export AGG_TO_ISP;
[edit policy-options]
+ policy-statement AGG_TO_ISP {
+   term ACCEPT_AGG {
+     from {
+       protocol aggregate;
+       route-filter 10.42.0.0/16 exact;
+     }
+     then accept;
+   }
+   term REJECT_OTHERS {
+     then reject;
+   }
+ }
```

```
[edit]
cjones@R1# commit and-quit
commit complete
Exiting configuration mode
```

5. R2 の集約ルートを作成します。

```
cjones@R2> configure
Entering configuration mode
```

```
[edit]
cjones@R2# set routing-options aggregate route 10.42.0.0/16
```

6. 照合で一致した集約ルートを受け入れて、それ以外のルートは拒否するよう、R2 のポリシーを設定します。reject ステートメントが設定されていない場合、暗黙的に Accept されることに注意してください。

```
[edit]
cjones@R2# edit policy-options policy-statement AGG_TO_ISP

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R2# set term ACCEPT_AGG from protocol aggregate

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R2# set term ACCEPT_AGG from route-filter 10.42.0.0/16 exact

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R2# set term ACCEPT_AGG then accept

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R2# set term REJECT_OTHERS then reject

[edit policy-options policy-statement AGG_TO_ISP]
cjones@R2# top
```

7. BGP 設定のグループまたは隣接機器の export ステートメントにポリシーを設定します。

```
[edit]
cjones@R2# set protocols bgp group ISP-B neighbor 172.18.0.1 export AGG_TO_ISP
```

8. R2 の変更内容を確認してコミットします。

```
[edit]
cjones@R2# show | compare
[edit routing-options]
+ aggregate {
+   route 10.42.0.0/16;
+ }
[edit protocols bgp group ISP-B neighbor 172.18.0.1]
+ export AGG_TO_ISP;
[edit policy-options]
+ policy-statement AGG_TO_ISP {
+   term ACCEPT_AGG {
+     from {
+       protocol aggregate;
+       route-filter 10.42.0.0/16 exact;
+     }
+     then accept;
+   }
+   term REJECT_OTHERS {
+     then reject;
+   }
+ }

[edit]
cjones@R2# commit and-quit
commit complete
```

Exiting configuration mode

### 集約プレフィックスが AS 111 と AS 222 に送信されたかどうか確認するには

1. R1 の Adj-RIB-Out テーブルをチェックします。

```
cjones@R1> show route advertising-protocol bgp 172.16.0.1
```

```
inet.0:11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 10.42.0.0/16          Self                                I
```

2. R2 の Adj-RIB-Out テーブルをチェックします。

```
cjones@R2> show route advertising-protocol bgp 172.18.0.1
```

```
inet.0:11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 10.42.0.0/16          Self                                I
```

### インバウンドトラフィックを優先して ISP-B 経由でネットワークに送信するには

Junos では、トラフィックを AS 内に送信するには、AS パスのプリペンドを使用します。具体的には、これまでに紹介したものと同一ポリシー構造を使用します。この例では、集約ルートを ISP-A にエクスポートするポリシーをすでに設定しているので、このポリシーにそのまま追加できます。

1. R1 のエクスポートポリシーを修正して、AS をプリペンドします。

```
cjones@R1> configure
```

Entering configuration mode

[edit]

```
cjones@R1# set policy-options policy-statement AGG_TO_ISP term ACCEPT_AGG then as-path-prepend "65001 65001"
```

[edit]

```
cjones@R1# commit and-quit
```

commit complete

Exiting configuration mode

### AS パスのプリペンド設定が適用されていることを確認するには

Junos では、Adj-RIB-Out テーブルをチェックするだけで、変更された AS パスを確認できます。このテーブルでは、ポリシーの適用後に BGP によってアドバタイズされたプレフィックスの変更内容を確認できます。

1. Adj-RIB-Out テーブルをチェックします。

```
cjones@R1> show route advertising-protocol bgp 172.16.0.1
```

```
inet.0:11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
* 10.42.0.0/16          Self                                65001 65001 [65001] I
```

### アウトバウンドトラフィックを優先してISP-B 経由でネットワークから送信するには

1. 繰り返しになりますが、ポリシーを使用して、BGP 経由で学習されるプレフィックスを変更できます。この例では、R2 のシンプルなインポートポリシーを使用して、ISP-B からの全ルートの Local Preference を上げることができます。R2 のポリシーを作成して、ISP-B から学習された全ルートの Local Preference を上げます。

```
cjones@R2> configure
Entering configuration mode
```

```
[edit]
cjones@R2# set policy-options policy-statement ISPB-LOCALPREF then local-preference
110
```

2. このポリシーを R2 の BGP グループまたは隣接機器に import として適用します。次に、コミットします。

```
[edit]
cjones@R2# set protocols bgp group ISP-B neighbor 172.18.0.1 import ISPB-LOCALPREF
```

```
[edit]
cjones@R2# commit and-quit
commit complete
Exiting configuration mode
```

### Local Preference の値が変更されたことを確認するには

1. R2 の RIB をチェックします。プレフィックスが R1 から学習されなくなったことに注意してください。これは、プレフィックスに対する R1 の最適なルートが R2 経由で学習され、R1 がそのルートを学習元に再アドバタイズしなくなったことによります。

```
cjones@R2> show route protocol bgp
```

```
inet.0:11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
111.111.111.0/24    *[BGP/170] 00:51:34, localpref 110
                   AS path:222 I
                   > to 172.18.0.1 via ge-0/0/0.0
```

2. R1 の RIB をチェックします。この時点の優先パスは R2 経由で ISP-B を宛先としていることを確認します。

```
cjones@R1> show route protocol bgp
```

```
inet.0:11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
111.111.111.0/24    *[BGP/170] 00:02:42, localpref 110, from 2.2.2.2
                   AS path:222 I
                   > to 10.42.0.2 via ge-0/0/1.0
                   [BGP/170] 01:09:57, localpref 100
                   AS path:111 I
                   > to 172.16.0.1 via ge-0/0/0.0
```

## 結論

この章では、IOS と Junos の両方で一般的な要素を共有するシンプルなネットワーク設定を設定して比較する方法について学習しました。Junos が IGP と BGP の両方でルーティング操作のほとんどをポリシー設定に大きく依存していることは明らかです。ただし、Junos の場合は、階層構造によるメリットがあります。Junos の設定は、C 言語のプログラミングとは異なります。IOS の設定ほど、難しくも複雑でもありません。ただし、機能面は優れています。

この章で学習したタスクは、ルーター、スイッチ、ファイアウォールなど、あらゆる Junos デバイスに応用できます。本書の最後で紹介している参考情報が役に立つ機会もあるでしょう。

本書では、タスクの比較から始まり、各種の設定に続いて、サンプルトポロジーの設定手順について説明しました。これで、Junos への移行に今すぐ着手できるだけの十分な知識を習得したはずですが、Junos への移行時に、ネットワークの基盤となる技術が変わってしまうわけではなく、本書で習得した知識によってスムーズに対応できるようになります。