

IOSでのBGPルート集約の設定(&R); XE

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[BGPルート集約に関する情報](#)

[設定](#)

[ネットワーク図](#)

[コンフィギュレーション](#)

[例 1](#)

[例 2](#)

[例 3](#)

[例 4](#)

[例 5](#)

[確認](#)

[トラブルシューティング](#)

[シナリオ 1](#)

[シナリオ 2](#)

はじめに

このドキュメントでは、オプションの引数を使用してBGPルート集約を設定する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- ボーダーゲートウェイプロトコル(BGP)の基本動作
- プレフィックスリスト
- ルートマップ

使用するコンポーネント

このドキュメントの情報は、Cisco IOS XEソフトウェアバージョン17.xに基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

BGPルート集約に関する情報

BGPルート集約を使用すると、複数の特定のルートを1つの集約ルート（集約ルート）に結合して、ルーティングテーブルのサイズとアドバタイズメントのオーバーヘッドを削減できます。

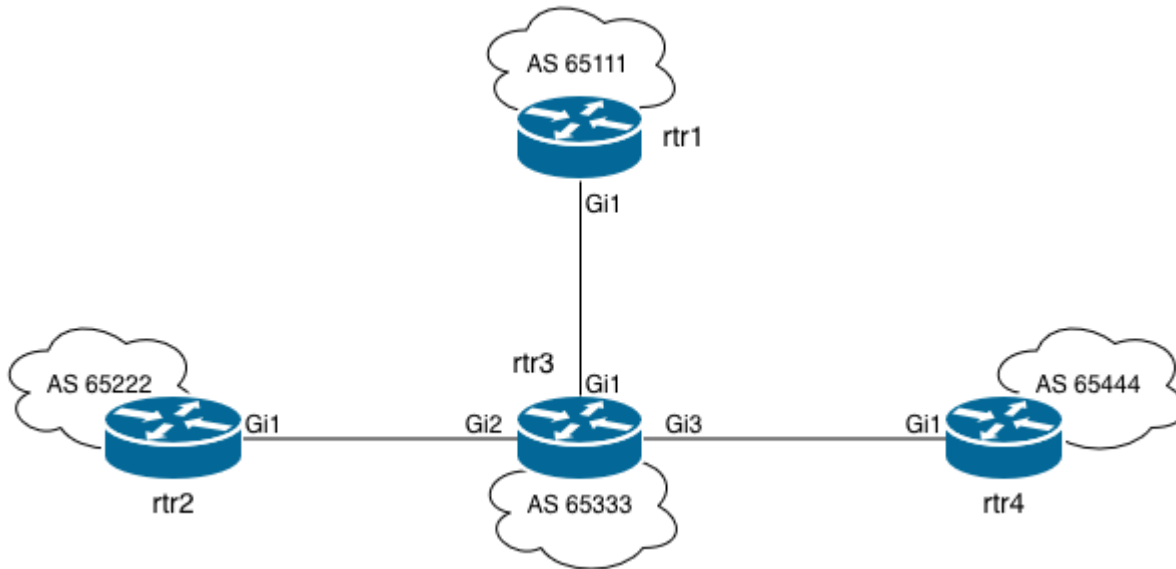
オプションのキーワードは次のとおりです。

- `as-set` : 集約ルートからのすべてのAS番号を含むASセットパスアトリビュートを生成します。
- `summary-only` : 集約経路だけをアドバタイズし、より具体的な経路を抑制します。
- `suppress-map <map-name>` : ルートマップに基づいて、選択されたより具体的なルートを抑制します。
- `advertise-map <map-name>` : ルートマップに基づいて、集約経路を条件付きでアドバタイズします。
- `attribute-map <map-name>` : ルートマップを使用して集約経路のアトリビュートを設定します。

デフォルトでは、`aggregate-address`コマンドは、ASパスを継承せずに、1つ以上の特定のルートが存在する場合にのみ集約アドレスを生成します。

設定

ネットワーク図



コンフィギュレーション

これはrtr3の初期設定です。

```
<#root>
```

```
rtr3#
```

```
show running-config | sec router bgp
```

```
router bgp 65333
bgp log-neighbor-changes
neighbor 10.13.13.1 remote-as 65111
neighbor 10.23.23.2 remote-as 65222
neighbor 10.34.34.4 remote-as 65444
```

これはrtr3のBGPテーブルです。

```
<#root>
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 9, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
```

RPKI validation codes: V valid, I invalid, N Not found

```
Network Next Hop Metric LocPrf Weight Path
*> 10.1.1.0/32 10.13.13.1 0 0 65111 i
*> 10.1.1.1/32 10.13.13.1 0 0 65111 i
*> 10.1.1.2/32 10.13.13.1 0 0 65111 i
*> 10.1.1.3/32 10.13.13.1 0 0 65111 i
*> 10.2.2.0/32 10.23.23.2 0 0 65222 i
*> 10.2.2.1/32 10.23.23.2 0 0 65222 i
*> 10.2.2.2/32 10.23.23.2 0 0 65222 i
*> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

rtr3がrtr1(AS 65111)およびrtr2(AS 65222)からの特定のプレフィックスをすべて保持していることに注目してください。rtr3はこれらのプレフィックスをrtr4にアドバタイズし、AS 65333をAS PATH属性に追加します。

rtr4は次の内容を受信します。

```
<#root>
```

```
rtr4#
```

```
show ip bgp
```

```
BGP table version is 9, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
*> 10.1.1.0/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.1/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.2/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.3/32 10.34.34.3 0 65333 65111 i
*> 10.2.2.0/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.1/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.2/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.3/32 10.34.34.3 0 65333 65222 i
```

例 1

集約アドレスだけをアドバタイズするようにBGPを設定します。

最初の実施例では、rtr4はプレフィクス10.0.0.0/8だけを受信します。

<#root>

```
rtr3(config)#router bgp 65333  
rtr3(config-router)#
```

```
aggregate-address 10.0.0.0 255.0.0.0 summary-only
```

```
rtr3(config-router)#exit  
rtr3(config)#
```

集約の作成後のrtr4のBGPテーブル :

<#root>

rtr4#

```
show ip bgp
```

```
BGP table version is 18, local router ID is 10.34.34.3  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path, L long-lived-stale,  
Origin codes: i - IGP, e - EGP, ? - incomplete  
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path  
*> 10.0.0.0 10.34.34.3 0 0 65333 i
```

ASパスが65333 (サマリーの発信元のルータ) であることに注目してください。

例 2

2番目の実施例では、元のAS-PATHを追跡する方法を説明します。

これはrtr3の設定です。

<#root>

```
rtr3#configure terminal  
rtr3(config)#router bgp 65333  
rtr3(config-router)#
```

```
aggregate-address 10.0.0.0 255.0.0.0 as-set summary-only
```

```
rtr3(config-router)#exit
```

これはrtr4のBGPテーブルです。

```
<#root>
```

```
rtr4#
```

```
show ip bgp
```

```
BGP table version is 36, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
```

```
*> 10.0.0.0 10.34.34.3 0 0 65333 {65111,65222} i
```

rtr3が集約しているすべてのASパスがあることに注意してください。

例 3

ルートマップを使用してBGPプレフィックスを抑制します。

3番目の実例では、rtr1から着信するプレフィックスをフィルタリングするルートマップを設定します。

```
<#root>
```

```
rtr3(config)#
```

```
ip prefix-list suppress_rtr1 permit 10.1.1.0/24 le 32
```

```
rtr3(config)#route-map
```

```
SUPPRESS-RTR1
```

```
permit 10
```

```
rtr3(config-route-map)#match ip address prefix-list
```

```
suppress_rtr1
```

```
rtr3(config)#  
rtr3(config)#router bgp 65333  
rtr3(config-router)#aggregate-address 10.0.0.0 255.0.0.0
```

```
suppress-map SUPPRESS-RTR1
```

```
rtr3(config-router)#end  
rtr3#
```

これはrtr3のBGPテーブルです。

```
<#root>
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 114, local router ID is 10.34.34.3  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path, L long-lived-stale,  
Origin codes: i - IGP, e - EGP, ? - incomplete  
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
```

```
*> 10.0.0.0 0.0.0.0 32768 i  
s> 10.1.1.0/32 10.13.13.1 0 0 65111 i  
s> 10.1.1.1/32 10.13.13.1 0 0 65111 i  
s> 10.1.1.2/32 10.13.13.1 0 0 65111 i  
s> 10.1.1.3/32 10.13.13.1 0 0 65111 i  
*> 10.2.2.0/32 10.23.23.2 0 0 65222 i  
*> 10.2.2.1/32 10.23.23.2 0 0 65222 i  
*> 10.2.2.2/32 10.23.23.2 0 0 65222 i  
*> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

rtr1からのプレフィックスだけが抑制されていることに注意してください。

これはrtr4のBGPテーブルです。

```
<#root>
```

```
rtr4#
```

```
show ip bgp
```

```
BGP table version is 114, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0 10.34.34.3 0 0 65333 i
*> 10.2.2.0/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.1/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.2/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.3/32 10.34.34.3 0 65333 65222 i
rtr4#
```

例 4

範囲内のプレフィクスがBGPテーブルに存在する場合にのみ、集約ルートをアドバタイズします。

4番目の例では、前に設定したのと同じルートマップを使用して、rtr1からのすべてのプレフィックスを抑制します。

```
rtr3(config)#router bgp 65333
rtr3(config-router)#aggregate-address 10.0.0.0 255.0.0.0 advertise-map SUPPRESS-RTR1 summary-only
```

advertise-mapによって条件が設定されます。BGPテーブルに10.1.1.0/24の範囲内のプレフィックスが存在する場合にのみ、サマリーのための集約が生成されます。

これはrtr3のBGPテーブルです。

```
<#root>
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 148, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
```

x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```
Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0 0.0.0.0 32768 i
s> 10.1.1.0/32 10.13.13.1 0 0 65111 i
s> 10.1.1.1/32 10.13.13.1 0 0 65111 i
s> 10.1.1.2/32 10.13.13.1 0 0 65111 i
s> 10.1.1.3/32 10.13.13.1 0 0 65111 i
s> 10.2.2.0/32 10.23.23.2 0 0 65222 i
s> 10.2.2.1/32 10.23.23.2 0 0 65222 i
s> 10.2.2.2/32 10.23.23.2 0 0 65222 i
s> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

ルートマップに一致するプレフィックスがない場合の出力を次に示します。

```
<#root>
```

```
rtr3#
```

```
show run | section router bgp
```

```
router bgp 65333
 aggregate-address 10.0.0.0 255.0.0.0 summary-only advertise-map SUPPRESS-RTR1
 neighbor 10.13.13.1 remote-as 65111
 neighbor 10.23.23.2 remote-as 65222
 neighbor 10.34.34.4 remote-as 65444
!
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 31, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
*> 10.2.2.0/32 10.23.23.2 0 0 65222 i
*> 10.2.2.1/32 10.23.23.2 0 0 65222 i
*> 10.2.2.2/32 10.23.23.2 0 0 65222 i
*> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

rtr2からのプレフィックスは抑制されておらず、集約ルートも生成されていないことに注意してください。

例 5

ルートマップを使用してBGPアトリビュートを設定します。

```
<#root>
rtr3(config)#route-map

BGP-ATTR

    permit 10
rtr3(config-route-map)#set community

no-export

rtr3(config-route-map)#exit
rtr3(config)#router bgp 65333
rtr3(config-router)#aggregate-address 10.0.0.0 255.0.0.0

attribute-map BGP-ATTR

    summary-only
```

生成された集約アドレスを確認すると、どのピアにもアドバタイズされていないことがわかります。すべてのネイバーが外部(eBGP)であり、no-export well-knownコミュニティを設定しているため、この特定のシナリオではこの状態が想定されます。summary-only引数を使用した結果、rtr4はルートを受信しません。

これはrtr3のBGPテーブルです

```
<#root>
rtr3#

show ip bgp 10.0.0.0
```

```
BGP routing table entry for 10.0.0.0/8, version 20
Paths: (1 available, best #1, table default, not advertised to EBGp peer)
Not advertised to any peer
Refresh Epoch 1
Local, (aggregated by 65333 10.34.34.3)
0.0.0.0 from 0.0.0.0 (10.34.34.3)
Origin IGP, localpref 100, weight 32768, valid, aggregated, local, atomic-aggregate, best
Community: no-export
rx pathid: 0, tx pathid: 0x0
Updated on Jun 12 2026 23:14:53 UTC
```

確認

BGPルート集約が正しく動作しているかどうかを確認するには、受信側ルータ（この例では rtr4）で、必要な情報だけを受信しているかどうかを確認します。たとえば、summary、summary、およびすべての特定のプレフィクス、summary、および一部の特定のプレフィクスのみです。次のコマンドを主に使用できます。

- show ip bgp
- show ip bgp route-map <マップ名>
- show running-config |セクションrouter bgp
- debug ip bgp update

詳細については、トラブルシューティングシナリオを参照してください。

トラブルシューティング

シナリオ 1

集約ルートが受信されず、特定のプレフィクスが表示される。

```
<#root>
```

```
rtr4#
```

```
show ip bgp
```

```
BGP table version is 9, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
*> 10.1.1.0/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.1/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.2/32 10.34.34.3 0 65333 65111 i
*> 10.1.1.3/32 10.34.34.3 0 65333 65111 i
*> 10.2.2.0/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.1/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.2/32 10.34.34.3 0 65333 65222 i
*> 10.2.2.3/32 10.34.34.3 0 65333 65222 i
```

summary-onlyがaggregate-addressの引数として設定されており、サブネットマスクが正しいことを確認します。

これはrtr3の設定です。

```
<#root>
```

```
rtr3(config)#router bgp 65333  
rtr3(config-router)#
```

```
aggregate-address 10.0.0.0 255.255.255.0 summary-only
```

```
rtr3(config-router)#exit  
rtr3(config)#
```

この例では、summary-addressが設定されていますが、サブネットマスクが正しくありません。10.0.0.0/24ネットワークのプレフィックスのみが集約に含まれるため、BGPルート集約のルールが破綻します。rtr1とrtr2からのすべてのプレフィックスはサマリー範囲から外れ、rtr3のBGPテーブルをチェックしても、抑制は行われません。

rtr3 : 修正された設定。

```
<#root>
```

```
rtr3(config)#router bgp 65333  
rtr3(config-router)#
```

```
aggregate-address 10.0.0.0 255.0.0.0 summary-only
```

```
rtr3(config-router)#exit  
rtr3(config)#
```

特定のプレフィックスに省略フラグが付いていることを確認します。

```
<#root>
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 18, local router ID is 10.34.34.3
```

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Network Next Hop Metric LocPrf Weight Path

```
*> 10.0.0.0 0.0.0.0 32768 i
s> 10.1.1.0/32 10.13.13.1 0 0 65111 i
s> 10.1.1.1/32 10.13.13.1 0 0 65111 i
s> 10.1.1.2/32 10.13.13.1 0 0 65111 i
s> 10.1.1.3/32 10.13.13.1 0 0 65111 i
s> 10.2.2.0/32 10.23.23.2 0 0 65222 i
s> 10.2.2.1/32 10.23.23.2 0 0 65222 i
s> 10.2.2.2/32 10.23.23.2 0 0 65222 i
s> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

シナリオ 2

[マップを省略]は、範囲を省略するように設定されていますが、省略されているものではありません。

rtr3のBGPテーブルについて考えます。すべての10.2.2.0/24プレフィックスを抑制する必要がありますが、設定を適用した後は機能しません。

<#root>

rtr3#

show ip bgp

```
BGP table version is 37, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network Next Hop Metric LocPrf Weight Path

```
*> 10.1.1.0/32 10.13.13.1 0 0 65111 i
*> 10.1.1.1/32 10.13.13.1 0 0 65111 i
*> 10.1.1.2/32 10.13.13.1 0 0 65111 i
*> 10.1.1.3/32 10.13.13.1 0 0 65111 i
*> 10.2.2.0/32 10.23.23.2 0 0 65222 i
*> 10.2.2.1/32 10.23.23.2 0 0 65222 i
*> 10.2.2.2/32 10.23.23.2 0 0 65222 i
*> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

rtr3のBGP設定を確認します。

```
<#root>
```

```
rtr3#
```

```
show run | section router bgp
```

```
router bgp 65333
 aggregate-address 10.0.0.0 255.0.0.0 suppress-map SUPPRESS-RTR2
 neighbor 10.13.13.1 remote-as 65111
 neighbor 10.23.23.2 remote-as 65222
 neighbor 10.34.34.4 remote-as 65444
rtr3#
```

設定したルートマップを確認します。

```
<#root>
```

```
rtr3#
```

```
show route-map SUPPRESS-RTR2
```

```
route-map SUPPRESS-RTR2, permit, sequence 10
Match clauses:
ip address prefix-lists:
```

```
suppress-rtr2
```

```
Set clauses:
Policy routing matches: 0 packets, 0 bytes
```

設定されているプレフィックスリストを確認します。

```
<#root>
```

```
rtr3#
```

```
show ip prefix-list suppress-rtr2
```

```
ip prefix-list suppress-rtr2: 1 entries
```

```
seq 5 permit 10.2.2.0/24
```

```
rtr3#
```

例のプレフィックスリストは10.2.2.0プレフィクスと正確に一致するため、より具体的なプレフィクスは抑制されません。より具体的なプレフィクスと一致させるには、より小さいか等しい演算子を使用する必要があります。

プレフィックスリストの設定を修正します。

```
<#root>
```

```
rtr3#configure terminal
rtr3(config)#no ip prefix-list suppress-rtr2
rtr3(config)#

ip prefix-list suppress-rtr2 permit 10.2.2.0/24 le 32

rtr3(config)#end
rtr3#
```

これは、prefix-list suppress-rtr2の正しい設定の後のrtr3 BGPテーブルです。

```
<#root>
```

```
rtr3#
```

```
show ip bgp
```

```
BGP table version is 14, local router ID is 10.34.34.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0 0.0.0.0 32768 i
*> 10.1.1.0/32 10.13.13.1 0 0 65111 i
*> 10.1.1.1/32 10.13.13.1 0 0 65111 i
*> 10.1.1.2/32 10.13.13.1 0 0 65111 i
*> 10.1.1.3/32 10.13.13.1 0 0 65111 i
s> 10.2.2.0/32 10.23.23.2 0 0 65222 i
s> 10.2.2.1/32 10.23.23.2 0 0 65222 i
s> 10.2.2.2/32 10.23.23.2 0 0 65222 i
s> 10.2.2.3/32 10.23.23.2 0 0 65222 i
```

BGPオプション引数を使用して集約経路を設定する方法を学習しました。また、経路集約が失敗する原因となる2つの一般的な例も示しました。設定で見つかる可能性のある最も一般的な誤りは次のとおりです。集約経路の生成に問題があり、問題のある設定を特定できない場合は、`debug ip bgp update`コマンドを使用して、より詳細な情報を収集します。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。