

# Configurazione e controllo di una rete di provider upstream con valori della community BGP

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Convenzioni](#)

[Configurazione e controllo dei criteri di routing](#)

[Esempio di rete](#)

[Configurazioni](#)

[R3](#)

[R1](#)

[R2](#)

[Verifica](#)

[Informazioni correlate](#)

## Introduzione

Questo documento descrive come usare i valori della community BGP per controllare la policy di routing nelle reti del provider upstream.

## Prerequisiti

### Requisiti

Questo documento richiede la comprensione del protocollo di routing BGP e del relativo funzionamento.

### Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware. Tuttavia, le informazioni di questo documento si basano sulle seguenti versioni software e hardware:

- Software Cisco IOS® versione 12.2(27)
- Cisco serie 2500 Router

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

# Premesse

Mentre le comunità stesse non alterano il processo [BGP Best Path](#), le comunità possono essere usate come flag per contrassegnare una serie di percorsi. I router di provider di servizi upstream possono quindi utilizzare questi flag per applicare criteri di routing specifici (ad esempio, la preferenza locale) all'interno della rete.

I provider eseguono il mapping tra i valori della community configurabili e i valori delle preferenze locali corrispondenti all'interno della rete del provider. È possibile disporre di criteri specifici che richiedono la modifica di LOCAL\_PREF nella rete del provider per impostare i valori della community corrispondenti nei relativi aggiornamenti di routing.

Una community è un gruppo di prefissi che condividono alcune proprietà comuni e che possono essere configurati con l'attributo community BGP. L'attributo della community BGP è un attributo transitivo facoltativo di lunghezza variabile. L'attributo è costituito da un set di quattro valori di ottetto che specificano una comunità. I valori degli attributi della community sono codificati con un numero AS (Autonomous System) nei primi due ottetti, mentre gli altri due ottetti sono definiti dall'AS. Un prefisso può avere più di un attributo di comunità. Un altoparlante BGP che vede più attributi della community in un prefisso può agire sulla base di uno, alcuni o tutti gli attributi. Un router può aggiungere o modificare un attributo della community prima di passarlo ad altri peer. Per ulteriori informazioni sull'attributo community, consultare i [casi di studio BGP](#).

L'attributo di preferenza locale indica all'AS il percorso preferito per raggiungere una determinata rete. Se esistono più percorsi alla stessa destinazione, viene scelto il percorso con la preferenza più alta (il valore predefinito dell'attributo di preferenza locale è 100). Per ulteriori informazioni, consultare i casi di studio .

## Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

## Configurazione e controllo dei criteri di routing

**Nota:** Per ulteriori informazioni sui comandi menzionati in questo documento, usare lo strumento di ricerca dei comandi.

Per semplificare le procedure, si presume che l'attributo community e l'attributo preferenza locale mappino tra il provider di servizi a monte (AS 100) e il dispositivo (AS 30).

Preferenza locale	Valori della community
130	100:300
125	100:250

Se i prefissi sono annunciati con un attributo community uguale a 100:300, il provider di servizi a monte imposta la preferenza locale di tali route su 130 e 125 se l'attributo community è uguale a 100:250.

In questo modo è possibile controllare i criteri di routing all'interno della rete del provider di servizi se si modificano i valori della community dei prefissi annunciati al provider di servizi.

Nel diagramma di rete , l'AS 30 desidera utilizzare questo criterio di routing con gli attributi della community.

- Il traffico in entrata da AS 100 destinato alla rete 10.0.10.0/24 attraversa il collegamento R1-R3. Se il collegamento R1-R3 ha esito negativo, tutto il traffico attraversa R2-R3.
- Il traffico in entrata da AS 100 destinato alla rete 10.1.0.0/24 attraversa il collegamento R2-R3. Se il collegamento R2-R3 ha esito negativo, tutto il traffico attraversa R1-R3.

Per ottenere questo criterio di routing, R3 annuncia i prefissi come segue:

A R1:

- 10.0.10.0/24 con attributo community 100:300
- 10.1.0.0/24 con attributo community 100:250

A R2:

- 10.0.10.0/24 con attributo community 100:250
- 10.1.0.0/24 con attributo community 100:300

Una volta che i router BGP adiacenti R1 e R2 ricevono i prefissi da R3, R1 e R2, applicano il criterio configurato in base al mapping tra gli attributi della community e delle preferenze locali (mostrato nella tabella precedente), in modo da ottenere il criterio di routing desiderato (AS 30). R1 installa i prefissi nella tabella BGP.

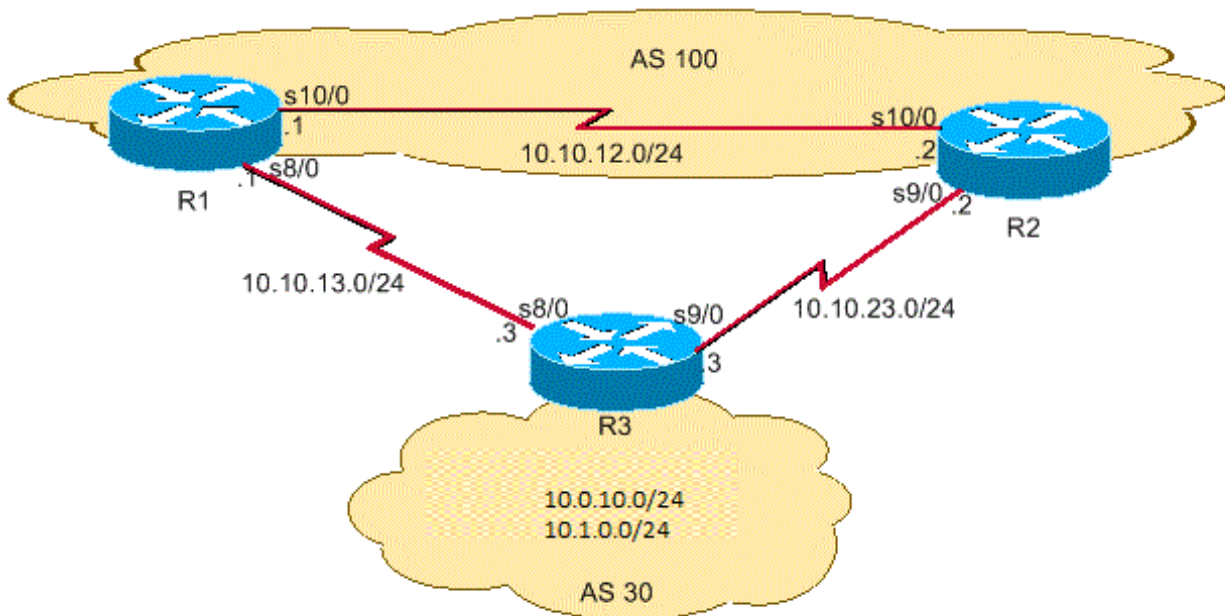
- 10.0.10.0/24 con una preferenza locale di 130
- 10.1.0.0/24 con una preferenza locale di 125

R2 installa il prefisso nella relativa tabella BGP:

- 10.0.10.0/24 con una preferenza locale di 125
- 10.1.0.0/24 con una preferenza locale di 130

Poiché nei criteri di selezione del percorso BGP si preferisce una preferenza locale maggiore, il percorso con preferenza locale pari a 130 (130 è maggiore di 125) viene selezionato come miglior percorso all'interno di AS 100 e installato nella tabella di routing IP di R1 e R2. Per ulteriori informazioni sui criteri di selezione del percorso BGP, fare riferimento all'[algoritmo di selezione del miglior percorso BGP](#).

## Esempio di rete



Reti BGP

## Configurazioni

Nel documento vengono usate queste configurazioni:

- R3
- R1
- R2

### R3

```
Current configuration : 2037 bytes !
version 12.2 !
hostname R3 !
interface Loopback0 ip address 10.0.10.0 255.255.255.0 !
interface Ethernet0/0 ip address 10.1.0.0 255.255.255.1 !
interface Serial8/0 ip address 10.10.13.3 255.255.255.0 !--- Interface connected to R1.
interface Serial9/0 ip address 10.10.23.3 255.255.255.0 !--- Interface connected to R2.
router bgp 30
network 10.0.10.0 mask 255.255.255.0
network 10.1.0.0 mask 255.255.255.1 !--- Network commands announce prefix 10.0.10.0/24 !--- and 10.1.0.0/24.
neighbor 10.10.13.1 remote-as 100 !--- Establishes peering with R1.
neighbor 10.10.13.1 send-community - !--- Without this command, the community attributes !--- are not sent to the neighbor.
neighbor 10.10.13.1 route-map Peer-R1 out !--- Configures outbound policy as defined by !--- route-map "Peer-R1" when peering with R1.
neighbor 10.10.23.2 remote-as 100 !--- Establishes peering with R2.
neighbor 10.10.23.2 send-community !--- Configures to send community attribute to R2.
neighbor 10.10.23.2 route-map Peer-R2 out !--- Configures outbound policy as defined by !--- route-map "Peer-R2" when peering with R2.
no auto-summary !
ip classless
ip bgp-community new-format !--- Allows you to configure the BGP community !--- attribute in AA:NN format.
! access-list 101 permit ip host 10.0.10.0 host 255.255.255.0
access-list 102 permit ip host 10.1.0.0 host 255.255.255.1 !
! route-map Peer-R1 permit 10 match ip address 101 set community 100:300 !--- Sets community 100:300 for routes matching access-list 101.
! route-map Peer-R1 permit 20 match ip address 102 set community 100:250 !--- Sets community 100:250 for routes matching access-list 102.
! route-map Peer-R2 permit 10 match ip address 101 set community 100:250 !--- Sets community 100:250 for routes matching access-list 101.
! route-map Peer-R2 permit 20 match ip address 102 set community 100:300 !--- Sets community 100:300 for routes matching access-list 102.
! end
```

### R1

```

Version 12.2 ! hostname R1 ! interface Loopback0 ip address 200.200.10.1 255.255.255.0 !
interface Serial8/0 ip address 10.10.13.1 255.255.255.1 !--- Connected to R3. ! interface
Serial10/0 ip address 10.10.12.1 255.255.255.0 !--- Connected to R2. ! router bgp 100 no
synchronization bgp log-neighbor-changes neighbor 10.10.12.2 remote-as 100 !--- Establishes
peering with R2. neighbor 10.10.12.2 next-hop-self neighbor 10.10.13.3 remote-as 30 !---
Establishes peering with R3. neighbor 10.10.13.3 route-map Peer-R3 in !--- Configures the
inbound policy as defined by !--- route-map "Peer-R3" when peering with R3. no auto-summary ! ip
bgp-community new-format !--- Allows you to configure the BGP community !--- attribute in AA:NN
format. ip community-list 1 permit 100:300 ip community-list 2 permit 100:250 !--- Defines
community list 1 and 2. ! route-map Peer-R3 permit 10 match community 1 set local-preference 130
!--- Sets local preference 130 for all routes !--- matching community list 1. ! route-map Peer-
R3 permit 20 match community 2 set local-preference 125 !--- Sets local preference 125 for all
routes !--- matching community list 2. ! route-map Peer-R3 permit 30 !--- Without this permit 30
statement, updates that do not !--- match the permit 10 or permit 20 statements are dropped. !
end

```

## R2

```

Version 12.2 ! hostname R2 ! interface Loopback0 ip address 10.0.10.0 255.255.255.0 ! interface
Serial9/0 ip address 10.10.23.2 255.255.255.1 !--- Connected to R3. ! interface Serial10/0 ip
address 10.10.12.2 255.255.255.0 !--- Connected to R1. ! router bgp 100 no synchronization bgp
log-neighbor-changes neighbor 10.10.12.1 remote-as 100 !--- Establishes iBGP peering with R1.
neighbor 10.10.12.1 next-hop-self neighbor 10.10.23.3 remote-as 30 !--- Establishes peering with
R3. neighbor 10.10.23.3 route-map Peer-R3 in !--- Configures inbound policy as defined by !---
route-map "Peer-R3" when peering with R3. no auto-summary ! ip bgp-community new-format !---
Allows you to configure the BGP community !--- attribute in AA:NN format. ! ip community-list 1
permit 100:300 ip community-list 2 permit 100:250 !--- Defines community list 1 and 2. ! route-
map Peer-R3 permit 10 match community 1 set local-preference 130 !--- Sets local preference 130
for all routes !--- matching community list 1. ! route-map Peer-R3 permit 20 match community 2
set local-preference 125 !--- Sets local preference 125 for all routes !--- matching community
list 2. ! route-map Peer-R3 permit 30 !--- Without this permit 30 statement, updates that do not
!--- match the permit 10 or permit 20 statements are dropped. ! end

```

## Verifica

R1 riceve i prefissi 10.0.10.0/24 e 10.1.0.0/24 con le community 100:300 e 100:250, come mostrato nel prossimo risultato del comando **show ip bgp**.

**Nota:** Una volta installate queste route nella tabella BGP in base al criterio configurato, ai prefissi con community 100:300 viene assegnata la preferenza locale 130 e ai prefissi con community 100:250 la preferenza locale 125.

```

R1# show ip bgp 10.0.10.0 BGP routing table entry for 10.0.10.0/24, version 2 Paths: (1
available, best #1, table Default-IP-Routing-Table) Advertised to non peer-group peers:
10.10.12.2 30 10.10.13.3 from 10.10.13.3 (10.0.10.0) Origin IGP, metric 0, localpref 130, valid,
external, best Community: 100:300 !--- Prefix 10.0.10.0/24 with community 100:300 received from
!--- 10.10.13.3 (R3) is assigned local preference 130.

```

```

R1# show ip bgp 10.1.0.0 BGP routing table entry for 10.1.0.0/24, version 4 Paths: (2 available,
best #1, table Default-IP-Routing-Table) Advertised to non peer-group peers: 10.10.13.3 30
10.10.12.2 from 10.10.12.2 (10.1.0.0) Origin IGP, metric 0, localpref 130, valid, internal, best
.0!--- Received prefix 10.1.0.0/24 over iBGP from 10.10.12.2 !--- (R2) with local preference
130. !--- (R2) with local preference 130.

```

```

30 10.10.13.3 from 10.10.13.3 (198.50.100.0) Origin IGP, metric 0, localpref 125, valid,
external Community: 100:250 !--- Prefix 10.1.0.0/24 with community 100:250 received from !---
10.10.13.3 (R3) is assigned local preference 125.

```

```
R1# show ip bgp BGP table version is 4, local router ID is 200.200.10.1 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path * > 10.0.10.0/24 10.10.13.3 0 130 0 30 i
*>i 10.1.0.0/24 10.10.12.2 0 130 0 30 i * 10.10.13.3 0 125 0 30 i
```

Il comando **show ip bgp** su R1 conferma che il miglior percorso selezionato su R1 è con preferenza locale (LocalPrf) = 130. Analogamente, R2 riceve i prefissi 10.0.10.0/24 e 10.1.0.0/24 con le community 100:250 e 100:300, come mostrato in grassetto in questo output del comando **show ip bgp**:

**Nota:** Una volta installate queste route nella tabella BGP, in base al criterio configurato, ai prefissi con community 100:300 viene assegnata la preferenza locale 130 e ai prefissi con community 100:250 la preferenza locale 125.

```
R2# show ip bgp 10.0.10.0 BGP routing table entry for 10.0.10.0/24, version 2 Paths: (2
available, best #2, table Default-IP-Routing-Table) Advertised to non peer-group peers:
10.10.23.3 30 10.10.23.3 from 10.10.23.3 (10.0.10.0) Origin IGP, metric 0, localpref 125, valid,
external Community: 100:250 !--- Prefix 10.0.10.0/24 with community 100:250 received from !---
10.10.23.3 (R3) is assigned local preference 125.
```

```
30 10.10.12.1 from 10.10.12.1 (200.200.10.1) Origin IGP, metric 0, localpref 130, valid,
internal, best !--- Received prefix 10.0.10.0/24 over iBGP from 10.10.12.1 !--- (R1) with local
preference 130.
```

```
R2# show ip bgp 10.0.10.0 BGP routing table entry for 10.0.10.0/24, version 3 Paths: (1
available, best #1, table Default-IP-Routing-Table) Advertised to non peer-group peers:
10.10.12.1 30 10.10.23.3 from 10.10.23.3 (10.0.10.0) Origin IGP, metric 0, localpref 130, valid,
external, best Community: 100:300 !--- Prefix 10.1.0.0/24 with community 100:300 received from
!--- 10.10.23.3 (R3) is assigned local preference 130.
```

```
R2# show ip bgp BGP table version is 3, local router ID is 192.168.50.1 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path * 10.0.10.0/24 10.10.23.3 0 125 0 30 i
*>i 10.10.12.1 0 130 0 30 i * > 10.1.0.0/24 10.10.23.3 0 130 0 30 i
```

L'output del comando **show ip bgp** su R2 conferma che il miglior percorso selezionato su R2 è con preferenza locale (localPrf) = 130. Il percorso IP con prefisso 10.0.10.0/24 preferisce che il collegamento R1-R3 esca da AS 100 verso AS 30. Il comando **show ip route** su R1 e R2 conferma questa preferenza.

```
R1# show ip route 10.0.10.0 Routing entry for 10.0.10.0/24 Known via "bgp 100", distance 20,
metric 0 Tag 30, type external Last update from 10.10.13.3 3d21h ago Routing Descriptor Blocks:
* 10.10.13.3, from 10.10.13.3, 3d21h ago Route metric is 0, traffic share count is 1 AS Hops 1
!--- On R1, the IP route to prefix 10.0.10.0/24 points !--- to next hop 10.10.13.3 which is R3
serial 8/0 !--- interface on the R1-R3 link.
```

```
R2# show ip route 10.1.0.0 Routing entry for 10.1.0.0/24 Known via "bgp 100", distance 200,
metric 0 Tag 30, type internal Last update from 10.10.12.1 3d21h ago Routing Descriptor Blocks:
* 10.10.12.1, from 10.10.12.1, 3d21h ago Route metric is 0, traffic share count is 1 AS Hops 1
!--- On R2, IP route to prefix 10.1.0.0/24 points !--- to next hop R1 (10.10.12.1) on its iBGP
link. !--- Thus traffic to network 10.1.0.0/24 from R2 !--- exits through R2-R1 and then R1-R3
link from !--- AS 100 towards AS 30.
```

La route IP per il prefisso 10.1.0.0/24 preferisce il collegamento R2-R3 per uscire da AS 100 verso AS 30. Il comando **show ip route** su R1 e R2 conferma questa preferenza.

```
R2# show ip route 10.1.0.0 Routing entry for 10.1.0.0/24 Known via "bgp 100", distance 20,
metric 0 Tag 30, type external Last update from 10.10.23.3 3d22h ago Routing Descriptor Blocks:
* 10.10.23.3, from 10.10.23.3, 3d22h ago Route metric is 0, traffic share count is 1 AS Hops 1
!--- On R2, IP route to prefix 10.1.0.0/24 points !--- to next hop 10.10.23.3 which is R3 serial
9/0 !--- interface on R2-R3 link.
```

```
R1# show ip route 10.1.0.0 Routing entry for 10.1.0.0/24 Known via "bgp 100", distance 200,
metric 0 Tag 30, type internal Last update from 10.10.12.2 3d22h ago Routing Descriptor Blocks:
* 10.10.12.2, from 10.10.12.2, 3d22h ago Route metric is 0, traffic share count is 1 AS Hops 1
!--- On R1, IP route to prefix 10.1.0.0/24 points !--- to next hop R2 (10.10.12.2) on its iBGP
link. !--- Thus traffic to network 10.1.0.0/24 from R1 !--- exits through R1-R2 and then R2-R3
link !--- from AS 100 towards AS 30.
```

Se un collegamento non riesce, ad esempio il collegamento R1-R3, tutto il traffico deve seguire il collegamento R2-R3. È possibile simulare il traffico se si chiude il collegamento tra R1 e R3.

```
R1# conf t Enter configuration commands, one per line. End with CNTL/Z. R1(config)#int s8/0
R1(config-if)#shut R1(config-if)# 3d22h: %BGP-5-ADJCHANGE: neighbor 10.10.13.3 Down Interface
flap 3d22h: %LINK-5-CHANGED: Interface Serial8/0, changed state to administratively down 3d22h:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial8/0, changed state to down
```

Si noti la tabella di routing IP per i prefissi 10.0.10.0/24 e 10.1.0.0/24 su R1 e R2. Utilizzare il collegamento R2-R3 per uscire da AS 100.

```
R1# show ip route 10.0.10.0 Routing entry for 10.0.10.0/24 Known via "bgp 100", distance 200,
metric 0 Tag 30, type internal Last update from 10.10.12.2 00:01:47 ago Routing Descriptor
Blocks: * 10.10.12.2, from 10.10.12.2, 00:01:47 ago Route metric is 0, traffic share count is 1
AS Hops 1
```

```
R1# show ip route 10.1.0.0 Routing entry for 10.1.0.0/24 Known via "bgp 100", distance 200,
metric 0 Tag 30, type internal Last update from 10.10.12.2 3d22h ago Routing Descriptor Blocks:
* 10.10.12.2, from 10.10.12.2, 3d22h ago Route metric is 0, traffic share count is 1 AS Hops 1
```

Questo output del comando **show** mostra che il percorso ai prefissi 10.0.10.0/24 e 10.1.0.0/24 punta all'hop successivo 10.10.12.2, (R2), come previsto. Esaminare la tabella di routing IP in R2 per verificare l'hop successivo dei prefissi 10.0.10.0/24 e 10.1.0.0/24. Per il corretto funzionamento del criterio configurato, l'hop successivo deve essere R3.

```
R2# show ip route 10.0.10.0 Routing entry for 10.0.10.0/24 Known via "bgp 100", distance 20,
metric 0 Tag 30, type external Last update from 10.10.23.3 00:04:10 ago Routing Descriptor
Blocks: * 10.10.23.3, from 10.10.23.3, 00:04:10 ago Route metric is 0, traffic share count is 1
AS Hops 1
```

```
R2# show ip route 10.1.0.0 Routing entry for 10.1.0.0/24 Known via "bgp 100", distance 20,
metric 0 Tag 30, type external Last update from 10.10.23.3 3d22h ago Routing Descriptor Blocks:
* 10.10.23.3, from 10.10.23.3, 3d22h ago Route metric is 0, traffic share count is 1 AS Hops 1
```

L'hop successivo 10.10.23.3 è l'interfaccia R3 Serial 9/0 sul collegamento R2-R3. Ciò conferma che il criterio configurato funziona come previsto.

## Informazioni correlate

- [RFC 1998](#)
- [Risoluzione dei problemi BGP](#)
- [BGP: Domande frequenti](#)
- [Condivisione del carico con BGP](#)
- [Documentazione e supporto tecnico](#)