|  |
| --- |
| **BGP**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#bgp) |

|  |
| --- |
| BGP is one of the Exterior Gateway Protocols and the de facto standard interdomain |

|  |
| --- |
| routing protocol. The latest BGP version is 4. BGP-4 is described in **RFC 1771** and |

|  |
| --- |
| updated by **RFC 4271**. **RFC 2858** adds multiprotocol support to BGP. |

|  |
| --- |
| VyOS makes use of FRR and we would like to thank them for their effort! |

**Basic Concepts**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#basic-concepts)

**Autonomous Systems**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#autonomous-systems)

From **[RFC 1930](https://datatracker.ietf.org/doc/html/rfc1930.html)**:

An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy.

Each AS has an identifying number associated with it called an ASN. This is a two octet value ranging in value from 1 to 65535. The AS numbers 64512 through 65535 are defined as private AS numbers. Private AS numbers must not be advertised on the global Internet. The 2-byte AS number range has been exhausted. 4-byte AS numbers are specified in **[RFC 6793](https://datatracker.ietf.org/doc/html/rfc6793.html)**, and provide a pool of 4294967296 AS numbers.

The ASN is one of the essential elements of BGP. BGP is a distance vector routing protocol, and the AS-Path framework provides distance vector metric and loop detection to BGP.

|  |
| --- |
| **set protocols bgp system-as <asn>** |

Set local ASN that this router represents. This is a a mandatory option!

**Address Families**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#address-families)

Multiprotocol extensions enable BGP to carry routing information for multiple network layer protocols. BGP supports an Address Family Identifier (AFI) for IPv4 and IPv6.

**Route Selection**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-selection)

The route selection process used by FRR’s BGP implementation uses the following decision criterion, starting at the top of the list and going towards the bottom until one of the factors can be used.

1. **Weight check**   
 Prefer higher local weight routes to lower routes.

2. **Local preference check**   
 Prefer higher local preference routes to lower.

3. **Local route check**   
 Prefer local routes (statics, aggregates, redistributed) to received routes.

4. **AS path length check**   
 Prefer shortest hop-count AS\_PATHs.

5. **Origin check**   
Prefer the lowest origin type route. That is, prefer IGP origin routes to EGP, to Incomplete routes.

6. **MED check**   
Where routes with a MED were received from the same AS, prefer the route with the lowest MED.

7. **External check**   
Prefer the route received from an external, eBGP peer over routes received from other types of peers.

8. **IGP cost check**   
 Prefer the route with the lower IGP cost.

9. **Multi-path check**   
 If multi-pathing is enabled, then check whether the routes not yet distinguished in

|  |  |
| --- | --- |
| preference may be considered equal. If bgp bestpath as-path multipath-relax is set, all   |  | | --- | |  | |

such routes are considered equal, otherwise routes received via iBGP with identical

AS\_PATHs or routes received from eBGP neighbours in the same AS are considered equal.

10. **Already-selected external check**

Where both routes were received from eBGP peers, then prefer the route which is

|  |  |  |
| --- | --- | --- |
| already selected. Note that this check is not applied if bgp bestpath compare-   |  | | --- | |  |  |  | | --- | |  |   routerid is configured. This check can prevent some cases of oscillation. |

11. **Router-ID check**

Prefer the route with the lowest *router-ID*. If the route has   
an *ORIGINATOR\_ID* attribute, through iBGP reflection, then that router ID is used, otherwise the *router-ID* of the peer the route was received from is used.

12. **Cluster-List length check**

The route with the shortest cluster-list length is used. The cluster-list reflects the iBGP reflection path the route has taken.

13. **Peer address**

Prefer the route received from the peer with the higher transport layer address, as a last-resort tie-breaker.

**Capability Negotiation**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#capability-negotiation)

When adding IPv6 routing information exchange feature to BGP. There were some   
proposals. IETF IDR adopted a proposal called Multiprotocol Extension for BGP. The specification is described in **[RFC 2283](https://datatracker.ietf.org/doc/html/rfc2283.html)**. The protocol does not define new protocols. It defines new attributes to existing BGP. When it is used exchanging IPv6 routing information it is called BGP-4+. When it is used for exchanging multicast routing information it is called MBGP.

*bgpd* supports Multiprotocol Extension for BGP. So if a remote peer supports the protocol, *bgpd* can exchange IPv6 and/or multicast routing information.

Traditional BGP did not have the feature to detect a remote peer’s capabilities, e.g. whether it can handle prefix types other than IPv4 unicast routes. This was a big problem using   
Multiprotocol Extension for BGP in an operational network. **[RFC 2842](https://datatracker.ietf.org/doc/html/rfc2842.html)** adopted a feature called Capability Negotiation. *bgpd* use this Capability Negotiation to detect the remote peer’s

capabilities. If a peer is only configured as an IPv4 unicast neighbor, *bgpd* does not send these Capability Negotiation packets (at least not unless other optional BGP features require capability negotiation).

By default, FRR will bring up peering with minimal common capability for the both sides. For example, if the local router has unicast and multicast capabilities and the remote router only has unicast capability the local router will establish the connection with unicast only capability. When there are no common capabilities, FRR sends Unsupported Capability error and then resets the connection.

**Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#configuration)

**BGP Router Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#bgp-router-configuration)

First of all you must configure BGP router with the ASN. The AS number is an identifier for the autonomous system. The BGP protocol uses the AS number for detecting whether the BGP connection is internal or external. VyOS does not have a special command to start the BGP process. The BGP process starts when the first neighbor is configured.

|  |
| --- |
| **set protocols bgp system-as <asn>** |

Set local autonomous system number that this router represents. This is a mandatory option!

**Peers Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#peers-configuration)

Defining Peers[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#defining-peers)

|  |
| --- |
| **set protocols bgp neighbor <address|interface> remote-as <nasn>** |

This command creates a new neighbor whose remote-as is <nasn>. The neighbor address can be an IPv4 address or an IPv6 address or an interface to use for the connection. The command is applicable for peer and peer group.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> remote-as internal** |

Create a peer as you would when you specify an ASN, except that if the peers ASN is different

|  |  |
| --- | --- |
| than mine as specified under the protocols bgp <asn> command the connection will be denied.   |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> remote-as external** |

Create a peer as you would when you specify an ASN, except that if the peers ASN is the same

|  |  |
| --- | --- |
| as mine as specified under the protocols bgp <asn> command the connection will be denied.   |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> local-role <role> [strict]** |

BGP roles are defined in RFC **[RFC 9234](https://datatracker.ietf.org/doc/html/rfc9234.html)** and provide an easy way to add route leak prevention, detection and mitigation. The local Role value is negotiated with the new BGP Role capability which has a built-in check of the corresponding value. In case of a mismatch the new OPEN Roles Mismatch Notification <2, 11> would be sent. The correct Role pairs are:

Provider - Customer

Peer - Peer

RS-Server - RS-Client

|  |  |
| --- | --- |
| If strict is set the BGP session won’t become established until the BGP neighbor sets local Role   |  | | --- | |  | |

on its side. This configuration parameter is defined in RFC **[RFC 9234](https://datatracker.ietf.org/doc/html/rfc9234.html)** and is used to enforce the corresponding configuration at your counter-parts side.

Routes that are sent from provider, rs-server, or the peer local-role (or if received by customer, rs-client, or the peer local-role) will be marked with a new Only to Customer (OTC) attribute.

Routes with this attribute can only be sent to your neighbor if your local-role is provider or rs-server. Routes with this attribute can be received only if your local-role is customer or rs-client.

In case of peer-peer relationship routes can be received only if OTC value is equal to your neighbor AS number.

All these rules with OTC will help to detect and mitigate route leaks and happen automatically if local-role is set.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> shutdown** |

This command disable the peer or peer group. To reenable the peer use the delete form of this command.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> description <text>** |

Set description of the peer or peer group.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> update-source <address| interface>** |

Specify the IPv4 source address to use for the BGP session to this neighbor, may be specified as either an IPv4 address directly or as an interface name.

Capability Negotiation[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#id2)

|  |
| --- |
| **set protocols bgp neighbor <address|interface> capability dynamic** |

This command would allow the dynamic update of capabilities over an established BGP session.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> capability extended-nexthop** |

Allow bgp to negotiate the extended-nexthop capability with it’s peer. If you are peering over a IPv6 Link-Local address then this capability is turned on automatically. If you are peering over a IPv6 Global Address then turning on this command will allow BGP to install IPv4 routes with IPv6 nexthops if you do not have IPv4 configured on interfaces.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> disable-capability-negotiation** |

Suppress sending Capability Negotiation as OPEN message optional parameter to the peer. This command only affects the peer is configured other than IPv4 unicast configuration.

When remote peer does not have capability negotiation feature, remote peer will not send any capabilities at all. In that case, bgp configures the peer with configured capabilities.

You may prefer locally configured capabilities more than the negotiated capabilities even though

|  |  |
| --- | --- |
| remote peer sends capabilities. If the peer is configured by override-capability , VyOS ignores   |  | | --- | |  | |

received capabilities then override negotiated capabilities with configured values.

Additionally you should keep in mind that this feature fundamentally disables the ability to use widely deployed BGP features. BGP unnumbered, hostname support, AS4, Addpath, Route Refresh, ORF, Dynamic Capabilities, and graceful restart.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> override-capability** |

This command allow override the result of Capability Negotiation with local configuration. Ignore remote peer’s capability value.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> strict-capability-match** |

This command forces strictly compare remote capabilities and local capabilities. If capabilities are different, send Unsupported Capability error then reset connection.

You may want to disable sending Capability Negotiation OPEN message optional parameter to

|  |  |  |
| --- | --- | --- |
| the peer when remote peer does not implement Capability Negotiation. Please use disable-   |  | | --- | |  |  |  | | --- | |  |   capability-negotiation command to disable the feature. |

Peer Parameters[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#peer-parameters)

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> allowas-in number <number>** |

This command accept incoming routes with AS path containing AS number with the same value as the current system AS. This is used when you want to use the same AS number in your sites, but you can’t connect them directly.

The number parameter (1-10) configures the amount of accepted occurences of the system AS number in AS path.

This command is only allowed for eBGP peers. It is not applicable for peer groups.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> as-override** |

This command override AS number of the originating router with the local AS number.

Usually this configuration is used in PEs (Provider Edge) to replace the incoming customer AS number so the connected CE ( Customer Edge) can use the same AS number as the other customer sites. This allows customers of the provider network to use the same AS number across their sites.

This command is only allowed for eBGP peers.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> attribute-unchanged <as-path|med|next-hop>** |

This command specifies attributes to be left unchanged for advertisements sent to a peer or peer group.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> maximum-prefix <number>** |

This command specifies a maximum number of prefixes we can receive from a given peer. If this number is exceeded, the BGP session will be destroyed. The number range is 1 to 4294967295.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> nexthop-self** |

This command forces the BGP speaker to report itself as the next hop for an advertised route it advertised to a neighbor.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> remove-private-as** |

This command removes the private ASN of routes that are advertised to the configured peer. It removes only private ASNs on routes advertised to EBGP peers.

If the AS-Path for the route has only private ASNs, the private ASNs are removed.

If the AS-Path for the route has a private ASN between public ASNs, it is assumed that this is a design choice, and the private ASN is not removed.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> soft-reconfiguration inbound** |

Changes in BGP policies require the BGP session to be cleared. Clearing has a large negative impact on network operations. Soft reconfiguration enables you to generate inbound updates from a neighbor, change and activate BGP policies without clearing the BGP session.

This command specifies that route updates received from this neighbor will be stored   
unmodified, regardless of the inbound policy. When inbound soft reconfiguration is enabled, the stored updates are processed by the new policy configuration to create new inbound updates.

|  |
| --- |
| **Note** |

|  |
| --- |
| Storage of route updates uses memory. If you enable soft reconfiguration inbound for multiple |

|  |
| --- |
| neighbors, the amount of memory used can become significant. |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> weight <number>** |

This command specifies a default weight value for the neighbor’s routes. The number range is 1 to 65535.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> advertisement-interval <seconds>** |

This command specifies the minimum route advertisement interval for the peer. The interval value is 0 to 600 seconds, with the default advertisement interval being 0.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> disable-connected-check** |

This command allows peerings between directly connected eBGP peers using loopback addresses without adjusting the default TTL of 1.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> disable-send-community <extended|standard>** |

This command specifies that the community attribute should not be sent in route updates to a peer. By default community attribute is sent.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> ebgp-multihop <number>** |

This command allows sessions to be established with eBGP neighbors when they are multiple hops away. When the neighbor is not directly connected and this knob is not enabled, the session will not establish. The number of hops range is 1 to 255. This command is mutually exclusive

|  |  |
| --- | --- |
| with ttl-security hops .   |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> local-as <asn> [no-prepend] [replace-as]** |

Specify an alternate AS for this BGP process when interacting with the specified peer or peer group. With no modifiers, the specified local-as is prepended to the received AS\_PATH when receiving routing updates from the peer, and prepended to the outgoing AS\_PATH (after the process local AS) when transmitting local routes to the peer.

|  |  |
| --- | --- |
| If the no-prepend attribute is specified, then the supplied local-as is not prepended to the   |  | | --- | |  | |

received AS\_PATH.

|  |  |
| --- | --- |
| If the replace-as attribute is specified, then only the supplied local-as is prepended to the   |  | | --- | |  | |

AS\_PATH when transmitting local-route updates to this peer.

|  |
| --- |
| **Note** |

|  |
| --- |
| This command is only allowed for eBGP peers. |
| **set protocols bgp neighbor <address|interface> passive** |

Configures the BGP speaker so that it only accepts inbound connections from, but does not initiate outbound connections to the peer or peer group.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> password <text>** |

This command specifies a MD5 password to be used with the tcp socket that is being used to connect to the remote peer.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> ttl-security hops <number>** |

This command enforces Generalized TTL Security Mechanism (GTSM), as specified in **[RFC 5082](https://datatracker.ietf.org/doc/html/rfc5082.html)**. With this command, only neighbors that are specified number of hops away will be allowed to become neighbors. The number of hops range is 1 to 254. This command is mutually

|  |  |
| --- | --- |
| exclusive with ebgp-multihop .   |  | | --- | |  | |

Peer Groups[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#peer-groups)

Peer groups are used to help improve scaling by generating the same update information to all members of a peer group. Note that this means that the routes generated by a member of a peer group will be sent back to that originating peer with the originator identifier attribute set to indicated the originating peer. All peers not associated with a specific peer group are treated as belonging to a default peer group, and will share updates.

|  |
| --- |
| **set protocols bgp peer-group <name>** |

This command defines a new peer group. You can specify to the group the same parameters that you can specify for specific neighbors.

|  |
| --- |
| **Note** |

If you apply a parameter to an individual neighbor IP address, you override the action defined for a peer group that includes that IP address.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> peer-group <name>** |

This command bind specific peer to peer group with a given name.

**Network Advertisement Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#network-advertisement-configuration)

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> network <prefix>** |

This command is used for advertising IPv4 or IPv6 networks.

|  |
| --- |
| **Note** |

|  |
| --- |
| By default, the BGP prefix is advertised even if it’s not present in the routing table. This |

|  |
| --- |
| behaviour differs from the implementation of some vendors. |
| **set protocols bgp parameters network-import-check** |

This configuration modifies the behavior of the network statement. If you have this configured the underlying network must exist in the routing table.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> default-originate [route-map <name>]** |

By default, VyOS does not advertise a default route (0.0.0.0/0) even if it is in routing table. When you want to announce default routes to the peer, use this command. Using optional

|  |  |
| --- | --- |
| argument route-map you can inject the default route to given neighbor only if the conditions in   |  | | --- | |  | |

the route map are met.

**Route Aggregation Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-aggregation-configuration)

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> aggregate-address <prefix>** |

This command specifies an aggregate address. The router will also announce longer-prefixes inside of the aggregate address.

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> aggregate-address <prefix> as-set** |

This command specifies an aggregate address with a mathematical set of autonomous systems. This command summarizes the AS\_PATH attributes of all the individual routes.

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> aggregate-address <prefix> summary-only** |

This command specifies an aggregate address and provides that longer-prefixes inside of the aggregate address are suppressed before sending BGP updates out to peers.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> unsuppress-map <name>** |

This command applies route-map to selectively unsuppress prefixes suppressed by summarisation.

**Redistribution Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#redistribution-configuration)

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> redistribute <route source>** |

This command redistributes routing information from the given route source to the BGP process. There are six modes available for route source: connected, kernel, ospf, rip, static, table.

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> redistribute <route source> metric <number>** |

This command specifies metric (MED) for redistributed routes. The metric range is 0 to 4294967295. There are six modes available for route source: connected, kernel, ospf, rip, static, table.

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> redistribute <route source> route-map <name>** |

This command allows to use route map to filter redistributed routes. There are six modes available for route source: connected, kernel, ospf, rip, static, table.

**General Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#general-configuration)

Common parameters[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#common-parameters)

|  |
| --- |
| **set protocols bgp parameters router-id <id>** |

This command specifies the router-ID. If router ID is not specified it will use the highest interface IP address.

|  |
| --- |
| **set protocols bgp address-family <ipv4-unicast|ipv6-unicast> maximum-paths <ebgp|ibgp> <number>** |

This command defines the maximum number of parallel routes that the BGP can support. In order for BGP to use the second path, the following attributes have to match: Weight, Local Preference, AS Path (both AS number and AS path length), Origin code, MED, IGP metric. Also, the next hop address for each path must be different.

|  |
| --- |
| **set protocols bgp parameters log-neighbor-changes** |

This command enable logging neighbor up/down changes and reset reason.

|  |
| --- |
| **set protocols bgp parameters no-client-to-client-reflection** |

This command disables route reflection between route reflector clients. By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection is not required. In this

|  |  |
| --- | --- |
| case, use the no-client-to-client-reflection command to disable client-to-client reflection.   |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp parameters no-fast-external-failover** |

Disable immediate session reset if peer’s connected link goes down.

|  |
| --- |
| **set protocols bgp listen range <prefix> peer-group <name>** |

This command is useful if one desires to loosen the requirement for BGP to have strictly defined neighbors. Specifically what is allowed is for the local router to listen to a range of IPv4 or IPv6 addresses defined by a prefix and to accept BGP open messages. When a TCP connection (and subsequently a BGP open message) from within this range tries to connect the local router then the local router will respond and connect with the parameters that are defined within the peer group. One must define a peer-group for each range that is listed. If no peer-group is defined then an error will keep you from committing the configuration.

|  |
| --- |
| **set protocols bgp listen limit <number>** |

This command goes hand in hand with the listen range command to limit the amount of BGP neighbors that are allowed to connect to the local router. The limit range is 1 to 5000.

|  |
| --- |
| **set protocols bgp parameters ebgp-requires-policy** |

This command changes the eBGP behavior of FRR. By default FRR enables **[RFC](https://datatracker.ietf.org/doc/html/rfc8212.html)**   
**[8212](https://datatracker.ietf.org/doc/html/rfc8212.html)** functionality which affects how eBGP routes are advertised, namely no routes are advertised across eBGP sessions without some sort of egress route-map/policy in place. In VyOS however we have this RFC functionality disabled by default so that we can preserve backwards compatibility with older versions of VyOS. With this option one can enable **[RFC](https://datatracker.ietf.org/doc/html/rfc8212.html)**   
**[8212](https://datatracker.ietf.org/doc/html/rfc8212.html)** functionality to operate.

Administrative Distance[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#administrative-distance)

|  |
| --- |
| **set protocols bgp parameters distance global <external|internal|local> <distance>** |

This command change distance value of BGP. The arguments are the distance values for external routes, internal routes and local routes respectively. The distance range is 1 to 255.

|  |
| --- |
| **set protocols bgp parameters distance prefix <subnet> distance <distance>** |

This command sets the administrative distance for a particular route. The distance range is 1 to 255.

|  |
| --- |
| **Note** |

|  |
| --- |
| Routes with a distance of 255 are effectively disabled and not installed into the kernel. |

Timers[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#timers)

|  |
| --- |
| **set protocols bgp timers holdtime <seconds>** |

This command specifies hold-time in seconds. The timer range is 4 to 65535. The default value is 180 second. If you set value to 0 VyOS will not hold routes.

|  |
| --- |
| **set protocols bgp timers keepalive <seconds>** |

This command specifies keep-alive time in seconds. The timer can range from 4 to 65535. The default value is 60 second.

Route Dampening[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-dampening)

When a route fails, a routing update is sent to withdraw the route from the network’s routing tables. When the route is re-enabled, the change in availability is also advertised. A route that continually fails and returns requires a great deal of network traffic to update the network about the route’s status.

Route dampening wich described in **[RFC 2439](https://datatracker.ietf.org/doc/html/rfc2439.html)** enables you to identify routes that repeatedly fail and return. If route dampening is enabled, an unstable route accumulates penalties each time the route fails and returns. If the accumulated penalties exceed a threshold, the route is no longer advertised. This is route suppression. Routes that have been suppressed are re-entered into the routing table only when the amount of their penalty falls below a threshold.

A penalty of 1000 is assessed each time the route fails. When the penalties reach a predefined threshold (suppress-value), the router stops advertising the route.

Once a route is assessed a penalty, the penalty is decreased by half each time a predefined amount of time elapses (half-life-time). When the accumulated penalties fall below a predefined threshold (reuse-value), the route is unsuppressed and added back into the BGP routing table.

No route is suppressed indefinitely. Maximum-suppress-time defines the maximum time a route can be suppressed before it is re-advertised.

|  |
| --- |
| **set protocols bgp parameters dampening half-life <minutes>** |

This command defines the amount of time in minutes after which a penalty is reduced by half. The timer range is 10 to 45 minutes.

|  |
| --- |
| **set protocols bgp parameters dampening re-use <seconds>** |

This command defines the accumulated penalty amount at which the route is re-advertised. The penalty range is 1 to 20000.

|  |
| --- |
| **set protocols bgp parameters dampening start-suppress-time <seconds>** |

This command defines the accumulated penalty amount at which the route is suppressed. The penalty range is 1 to 20000.

|  |
| --- |
| **set protocols bgp parameters dampening max-suppress-time <seconds>** |

This command defines the maximum time in minutes that a route is suppressed. The timer range is 1 to 255 minutes.

**Route Selection Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-selection-configuration)

|  |
| --- |
| **set protocols bgp parameters always-compare-med** |

This command provides to compare the MED on routes, even when they were received from different neighbouring ASes. Setting this option makes the order of preference of routes more defined, and should eliminate MED induced oscillations.

|  |
| --- |
| **set protocols bgp parameters bestpath as-path confed** |

This command specifies that the length of confederation path sets and sequences should be taken into account during the BGP best path decision process.

|  |
| --- |
| **set protocols bgp parameters bestpath as-path multipath-relax** |

This command specifies that BGP decision process should consider paths of equal AS\_PATH length candidates for multipath computation. Without the knob, the entire AS\_PATH must match for multipath computation.

|  |
| --- |
| **set protocols bgp parameters bestpath as-path ignore** |

Ignore AS\_PATH length when selecting a route

|  |
| --- |
| **set protocols bgp parameters bestpath compare-routerid** |

Ensure that when comparing routes where both are equal on most metrics, including local-pref, AS\_PATH length, IGP cost, MED, that the tie is broken based on router-ID.

If this option is enabled, then the already-selected check, where already selected eBGP routes are preferred, is skipped.

If a route has an ORIGINATOR\_ID attribute because it has been reflected, that   
ORIGINATOR\_ID will be used. Otherwise, the router-ID of the peer the route was received from will be used.

The advantage of this is that the route-selection (at this point) will be more deterministic. The disadvantage is that a few or even one lowest-ID router may attract all traffic to otherwise-equal paths because of this check. It may increase the possibility of MED or IGP oscillation, unless other measures were taken to avoid these. The exact behaviour will be sensitive to the iBGP and reflection topology.

|  |
| --- |
| **set protocols bgp parameters bestpath med confed** |

This command specifies that BGP considers the MED when comparing routes originated from different sub-ASs within the confederation to which this BGP speaker belongs. The default state, where the MED attribute is not considered.

|  |
| --- |
| **set protocols bgp parameters bestpath med missing-as-worst** |

This command specifies that a route with a MED is always considered to be better than a route without a MED by causing the missing MED attribute to have a value of infinity. The default state, where the missing MED attribute is considered to have a value of zero.

|  |
| --- |
| **set protocols bgp parameters default local-pref <local-pref value>** |

This command specifies the default local preference value. The local preference range is 0 to 4294967295.

|  |
| --- |
| **set protocols bgp parameters deterministic-med** |

This command provides to compare different MED values that advertised by neighbours in the same AS for routes selection. When this command is enabled, routes from the same autonomous system are grouped together, and the best entries of each group are compared.

|  |
| --- |
| **set protocols bgp address-family ipv4-unicast network <prefix> backdoor** |

This command allows the router to prefer route to specified prefix learned via IGP through backdoor link instead of a route to the same prefix learned via EBGP.

**Route Filtering Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-filtering-configuration)

In order to control and modify routing information that is exchanged between peers you can use route-map, filter-list, prefix-list, distribute-list.

For inbound updates the order of preference is:

 route-map

 filter-list

 prefix-list, distribute-list   
For outbound updates the order of preference is:

 prefix-list, distribute-list

 filter-list

 route-map

|  |
| --- |
| **Note** |
| The attributes prefix-list and distribute-list are mutually exclusive, and only one command   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| (distribute-list or prefix-list) can be applied to each inbound or outbound direction for a |

|  |
| --- |
| particular neighbor. |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> distribute-list <export|import> <number>** |

This command applies the access list filters named in <number> to the specified BGP neighbor to restrict the routing information that BGP learns and/or advertises. The

|  |  |  |
| --- | --- | --- |
| arguments export and import specify the direction in which the access list are applied.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> prefix-list <export|import> <name>** |

This command applies the prfefix list filters named in <name> to the specified BGP neighbor to restrict the routing information that BGP learns and/or advertises. The

|  |  |  |
| --- | --- | --- |
| arguments export and import specify the direction in which the prefix list are applied.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> route-map <export|import> <name>** |

This command applies the route map named in <name> to the specified BGP neighbor to control and modify routing information that is exchanged between peers. The

|  |  |  |
| --- | --- | --- |
| arguments export and import specify the direction in which the route map are applied.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> filter-list <export|import> <name>** |

This command applies the AS path access list filters named in <name> to the specified BGP neighbor to restrict the routing information that BGP learns and/or advertises. The

|  |  |  |
| --- | --- | --- |
| arguments export and import specify the direction in which the AS path access list are applied.   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| **set protocols bgp neighbor <address|interface> address-family <ipv4-unicast|ipv6-unicast> capability orf <receive|send>** |

This command enables the ORF capability (described in **[RFC 5291](https://datatracker.ietf.org/doc/html/rfc5291.html)**) on the local router, and

|  |  |  |
| --- | --- | --- |
| enables ORF capability advertisement to the specified BGP peer. The receive keyword   |  | | --- | |  |   configures a router to advertise ORF receive capabilities. The send keyword configures a router   |  | | --- | |  | |

to advertise ORF send capabilities. To advertise a filter from a sender, you must create an IP prefix list for the specified BGP peer applied in inbound derection.

|  |
| --- |
| **set protocols bgp neighbor <address|interface> solo** |

This command prevents from sending back prefixes learned from the neighbor.

**BGP Scaling Configuration**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#bgp-scaling-configuration)

BGP routers connected inside the same AS through BGP belong to an internal BGP session, or IBGP. In order to prevent routing table loops, IBGP speaker does not advertise IBGP-learned routes to other IBGP speaker (Split Horizon mechanism). As such, IBGP requires a full mesh of all peers. For large networks, this quickly becomes unscalable.

There are two ways that help us to mitigate the BGPs full-mesh requirement in a network:

 Using BGP route-reflectors

 Using BGP confederation   
Route Reflector Configuration[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-reflector-configuration)

Introducing route reflectors removes the need for the full-mesh. When you configure a route reflector you have to tell the router whether the other IBGP router is a client or non-client. A client is an IBGP router that the route reflector will “reflect” routes [to, the non](https://datatracker.ietf.org/doc/html/rfc4456.html)-client is just a regular IBGP neighbor. Route reflectors mechanism is described in **[RFC 4456](https://datatracker.ietf.org/doc/html/rfc4456.html)** and updated by **[RFC 7606](https://datatracker.ietf.org/doc/html/rfc7606.html)**.

|  |
| --- |
| **set protocols bgp neighbor <address> address-family <ipv4-unicast|ipv6-unicast> route-reflector-client** |

This command specifies the given neighbor as route reflector client.

|  |
| --- |
| **set protocols bgp parameters cluster-id <id>** |

This command specifies cluster ID which identifies a collection of route reflectors and their clients, and is used by route reflectors to avoid looping. By default cluster ID is set to the BGP router id value, but can be set to an arbitrary 32-bit value.

Confederation Configuration[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#confederation-configuration)

A BGP confederation divides our AS into sub-ASes to reduce the number of required IBGP peerings. Within a sub-AS we still require full-mesh IBGP but between these sub-ASes we use something that looks like EBGP but behaves like IBGP (called confederation BGP).

Confederation mechanism is described in **[RFC 5065](https://datatracker.ietf.org/doc/html/rfc5065.html)**

|  |
| --- |
| **set protocols bgp parameters confederation identifier <asn>** |

This command specifies a BGP confederation identifier. <asn> is the number of the autonomous system that internally includes multiple sub-autonomous systems (a confederation).

|  |
| --- |
| **set protocols bgp parameters confederation peers <nsubasn>** |

This command sets other confederations <nsubasn> as members of autonomous system specified

|  |  |
| --- | --- |
| by confederation identifier <asn> .   |  | | --- | |  | |

**Operational Mode Commands**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#operational-mode-commands)

**Show**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#show)

|  |
| --- |
| **show <ip|ipv6> bgp** |

This command displays all entries in BGP routing table.

|  |
| --- |
| BGP table version is 10, local router ID is 10.0.35.3, vrf id 0  Default local pref 100, local AS 65000  Status codes: s suppressed, d damped, h history, \* valid, > best, = multipath, i internal, r RIB-failure, S Stale, R Removed  Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self  Origin codes: i - IGP, e - EGP, ? - incomplete |

|  |
| --- |
| Network Next Hop Metric LocPrf Weight Path \*> 198.51.100.0/24 10.0.34.4 0 0 65004 i \*> 203.0.113.0/24 10.0.35.5 0 0 65005 i |

|  |
| --- |
| Displayed 2 routes and 2 total paths |

|  |
| --- |
| **show <ip|ipv6> bgp <address|prefix>** |

This command displays information about the particular entry in the BGP routing table.

|  |
| --- |
| BGP routing table entry for 198.51.100.0/24  Paths: (1 available, best #1, table default) |

|  |
| --- |
| Advertised to non peer-group peers:  10.0.13.1 10.0.23.2 10.0.34.4 10.0.35.5  65004  10.0.34.4 from 10.0.34.4 (10.0.34.4)  Origin IGP, metric 0, valid, external, best (First path received) Last update: Wed Jan 6 12:18:53 2021 |
| **show ip bgp cidr-only** |

This command displays routes with classless interdomain routing (CIDR).

|  |
| --- |
| **show <ip|ipv6> bgp community <value>** |

This command displays routes that belong to specified BGP communities. Valid value is a community number in the range from 1 to 4294967200, or AA:NN (autonomous system-community number/2-byte number), no-export, local-as, or no-advertise.

|  |
| --- |
| **show <ip|ipv6> bgp community-list <name>** |

This command displays routes that are permitted by the BGP community list.

|  |
| --- |
| **show ip bgp dampened-paths** |

This command displays BGP dampened routes.

|  |
| --- |
| **show ip bgp flap-statistics** |

This command displays information about flapping BGP routes.

|  |
| --- |
| **show ip bgp filter-list <name>** |

This command displays BGP routes allowed by the specified AS Path access list.

|  |
| --- |
| **show <ip|ipv6> bgp neighbors <address> advertised-routes** |

This command displays BGP routes advertised to a neighbor.

|  |
| --- |
| **show <ip|ipv6> bgp neighbors <address> received-routes** |

This command displays BGP routes originating from the specified BGP neighbor before inbound policy is applied. To use this command inbound soft reconfiguration must be enabled.

|  |
| --- |
| **show <ip|ipv6> bgp neighbors <address> routes** |

This command displays BGP received-routes that are accepted after filtering.

|  |
| --- |
| **show <ip|ipv6> bgp neighbors <address> dampened-routes** |

This command displays dampened routes received from BGP neighbor.

|  |
| --- |
| **show <ip|ipv6> bgp regexp <text>** |

This command displays information about BGP routes whose AS path matches the specified regular expression.

|  |
| --- |
| **show <ip|ipv6> bgp summary** |

This command displays the status of all BGP connections.

|  |
| --- |
| IPv4 Unicast Summary:  BGP router identifier 10.0.35.3, local AS number 65000 vrf-id 0 BGP table version 11  RIB entries 5, using 920 bytes of memory  Peers 4, using 82 KiB of memory |

|  |
| --- |
| Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd 10.0.13.1 4 65000 148 159 0 0 0 02:16:01 0  10.0.23.2 4 65000 136 143 0 0 0 02:13:21 0  10.0.34.4 4 65004 161 163 0 0 0 02:16:01 1  10.0.35.5 4 65005 162 166 0 0 0 02:16:01 1 |

|  |
| --- |
| Total number of neighbors 4 |

**Reset**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#reset)

|  |
| --- |
| **reset <ip|ipv6> bgp <address> [soft [in|out]]** |

This command resets BGP connections to the specified neighbor IP address. With

|  |  |  |  |
| --- | --- | --- | --- |
| argument soft this command initiates a soft reset. If you do not specify the in or out options,   |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | |

both inbound and outbound soft reconfiguration are triggered.

|  |
| --- |
| **reset ip bgp all** |

This command resets all BGP connections of given router.

|  |
| --- |
| **reset ip bgp dampening** |

This command uses to clear BGP route dampening information and to unsuppress suppressed routes.

|  |
| --- |
| **reset ip bgp external** |

This command resets all external BGP peers of given router.

|  |
| --- |
| **reset ip bgp peer-group <name> [soft [in|out]]** |
| This command resets BGP connections to the specified peer group. With argument soft this   |  | | --- | |  |   command initiates a soft reset. If you do not specify the in or out options, both inbound and   |  | | --- | |  |  |  | | --- | |  | |

outbound soft reconfiguration are triggered.   
**Examples**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#examples)

**IPv4 peering**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#ipv4-peering)  
A simple eBGP configuration:

**Node 1:**

|  |
| --- |
| set protocols bgp system-as 65534  set protocols bgp neighbor 192.168.0.2 ebgp-multihop '2'  set protocols bgp neighbor 192.168.0.2 remote-as '65535'  set protocols bgp neighbor 192.168.0.2 update-source '192.168.0.1' set protocols bgp address-family ipv4-unicast network '172.16.0.0/16' set protocols bgp parameters router-id '192.168.0.1' |

**Node 2:**

|  |
| --- |
| set protocols bgp system-as 65535  set protocols bgp neighbor 192.168.0.1 ebgp-multihop '2'  set protocols bgp neighbor 192.168.0.1 remote-as '65534'  set protocols bgp neighbor 192.168.0.1 update-source '192.168.0.2' set protocols bgp address-family ipv4-unicast network '172.17.0.0/16' set protocols bgp parameters router-id '192.168.0.2' |

Don’t forget, the CIDR declared in the network statement MUST **exist in your routing table**

**(dynamic or static), the best way to make sure that is true is creating a static route:**

**Node 1:**

|  |
| --- |
| set protocols static route 172.16.0.0/16 blackhole distance '254' |

**Node 2:**

|  |
| --- |
| set protocols static route 172.17.0.0/16 blackhole distance '254' |

**IPv6 peering**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#ipv6-peering)  
A simple BGP configuration via IPv6.

**Node 1:**

|  |
| --- |
| set protocols bgp system-as 65534  set protocols bgp neighbor 2001:db8::2 ebgp-multihop '2'  set protocols bgp neighbor 2001:db8::2 remote-as '65535'  set protocols bgp neighbor 2001:db8::2 update-source '2001:db8::1' set protocols bgp neighbor 2001:db8::2 address-family ipv6-unicast set protocols bgp address-family ipv6-unicast network '2001:db8:1::/48' set protocols bgp parameters router-id '10.1.1.1' |

**Node 2:**

|  |
| --- |
| set protocols bgp system-as 65535 |

|  |
| --- |
| set protocols bgp neighbor 2001:db8::1 ebgp-multihop '2'  set protocols bgp neighbor 2001:db8::1 remote-as '65534'  set protocols bgp neighbor 2001:db8::1 update-source '2001:db8::2' set protocols bgp neighbor 2001:db8::1 address-family ipv6-unicast set protocols bgp address-family ipv6-unicast network '2001:db8:2::/48' set protocols bgp parameters router-id '10.1.1.2' |

Don’t forget, the CIDR declared in the network statement **MUST exist in your routing table**

**(dynamic or static), the best way to make sure that is true is creating a static route:**

**Node 1:**

|  |
| --- |
| set protocols static route6 2001:db8:1::/48 blackhole distance '254' |

**Node 2:**

|  |
| --- |
| set protocols static route6 2001:db8:2::/48 blackhole distance '254' |

**Route Filtering**[](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#route-filtering)  
Route filter can be applied using a route-map:

**Node1:**

|  |
| --- |
| set policy prefix-list AS65535-IN rule 10 action 'permit'  set policy prefix-list AS65535-IN rule 10 prefix '172.16.0.0/16' set policy prefix-list AS65535-OUT rule 10 action 'deny'  set policy prefix-list AS65535-OUT rule 10 prefix '172.16.0.0/16' set policy prefix-list6 AS65535-IN rule 10 action 'permit'  set policy prefix-list6 AS65535-IN rule 10 prefix '2001:db8:2::/48' set policy prefix-list6 AS65535-OUT rule 10 action 'deny'  set policy prefix-list6 AS65535-OUT rule 10 prefix '2001:db8:2::/48' |

|  |
| --- |
| set policy route-map AS65535-IN rule 10 action 'permit'  set policy route-map AS65535-IN rule 10 match ip address prefix-list 'AS65535-IN' set policy route-map AS65535-IN rule 10 match ipv6 address prefix-list 'AS65535-IN' set policy route-map AS65535-IN rule 20 action 'deny'  set policy route-map AS65535-OUT rule 10 action 'deny'  set policy route-map AS65535-OUT rule 10 match ip address prefix-list 'AS65535-OUT' set policy route-map AS65535-OUT rule 10 match ipv6 address prefix-list 'AS65535-OUT' set policy route-map AS65535-OUT rule 20 action 'permit' |

|  |
| --- |
| set protocols bgp system-as 65534  set protocols bgp neighbor 2001:db8::2 address-family ipv4-unicast route-map export 'AS65535-OUT' set protocols bgp neighbor 2001:db8::2 address-family ipv4-unicast route-map import 'AS65535-IN' set protocols bgp neighbor 2001:db8::2 address-family ipv6-unicast route-map export 'AS65535-OUT' set protocols bgp neighbor 2001:db8::2 address-family ipv6-unicast route-map import 'AS65535-IN' |

**Node2:**

|  |
| --- |
| set policy prefix-list AS65534-IN rule 10 action 'permit'  set policy prefix-list AS65534-IN rule 10 prefix '172.17.0.0/16' set policy prefix-list AS65534-OUT rule 10 action 'deny'  set policy prefix-list AS65534-OUT rule 10 prefix '172.17.0.0/16' set policy prefix-list6 AS65534-IN rule 10 action 'permit' |

|  |
| --- |
| set policy prefix-list6 AS65534-IN rule 10 prefix '2001:db8:1::/48' set policy prefix-list6 AS65534-OUT rule 10 action 'deny'  set policy prefix-list6 AS65534-OUT rule 10 prefix '2001:db8:1::/48' |

|  |
| --- |
| set policy route-map AS65534-IN rule 10 action 'permit'  set policy route-map AS65534-IN rule 10 match ip address prefix-list 'AS65534-IN' set policy route-map AS65534-IN rule 10 match ipv6 address prefix-list 'AS65534-IN' set policy route-map AS65534-IN rule 20 action 'deny'  set policy route-map AS65534-OUT rule 10 action 'deny'  set policy route-map AS65534-OUT rule 10 match ip address prefix-list 'AS65534-OUT' set policy route-map AS65534-OUT rule 10 match ipv6 address prefix-list 'AS65534-OUT' set policy route-map AS65534-OUT rule 20 action 'permit' |

|  |
| --- |
| set protocols bgp system-as 65535  set protocols bgp neighbor 2001:db8::1 address-family ipv4-unicast route-map export 'AS65534-OUT' set protocols bgp neighbor 2001:db8::1 address-family ipv4-unicast route-map import 'AS65534-IN' set protocols bgp neighbor 2001:db8::1 address-family ipv6-unicast route-map export 'AS65534-OUT' set protocols bgp neighbor 2001:db8::1 address-family ipv6-unicast route-map import 'AS65534-IN' |

We could expand on this and also deny link local and multicast in the rule 20 action deny.