**VRF**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf)

VRF devices combined with ip rules provides the ability to create virtual routing and forwarding domains (aka VRFs, VRF-lite to be specific) in the Linux network stack. One use case is the multi-tenancy problem where each tenant has their own unique routing tables and in the very least need different default gateways.

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

A VRF device is created with an associated route table. Network interfaces are then enslaved to a VRF device.

|  |
| --- |
| **set vrf name <name>** |

Create new VRF instance with *<name>*. The name is used when placing individual interfaces into the VRF.

|  |
| --- |
| **set vrf name <name> table <id>** |

Configured routing table *<id>* is used by VRF *<name>*.

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

|  |
| --- |
| A routing table ID can not be modified once it is assigned. It can only be changed by deleting |

|  |
| --- |
| and re-adding the VRF instance. |
| **set vrf bind-to-all** |

By default the scope of the port bindings for unbound sockets is limited to the default VRF. That is, it will not be matched by packets arriving on interfaces enslaved to a VRF and processes may bind to the same port if they bind to a VRF.

TCP & UDP services running in the default VRF context (ie., not bound to any VRF device) can work across all VRF domains by enabling this option.

**Zebra/Kernel route filtering**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#zebra-kernel-route-filtering)

Zebra supports prefix-lists and Route Mapss to match routes received from other FRR   
components. The permit/deny facilities provided by these commands can be used to filter which routes zebra will install in the kernel.

|  |
| --- |
| **set vrf <name> ip protocol <protocol> route-map <route-map>** |

Apply a route-map filter to routes for the specified protocol.

The following protocols can be used: any, babel, bgp, connected, eigrp, isis, kernel, ospf, rip, static, table

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

|  |
| --- |
| If you choose any as the option that will cause all protocols that are sending routes to zebra. |
| **set vrf <name> ipv6 protocol <protocol> route-map <route-map>** |

Apply a route-map filter to routes for the specified protocol.

The following protocols can be used: any, babel, bgp, connected, isis, kernel, ospfv3, ripng, static, table

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

|  |
| --- |
| If you choose any as the option that will cause all protocols that are sending routes to zebra. |

**Interfaces**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#interfaces)

When VRFs are used it is not only mandatory to create a VRF but also the VRF itself needs to be assigned to an interface.

|  |
| --- |
| **set interfaces <dummy | ethernet | bonding | bridge | pppoe> <interface> vrf <name>** |

Assign interface identified by *<interface>* to VRF named *<name>*.

**Routing**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#routing)

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

|  |
| --- |
| VyOS 1.4 (sagitta) introduced dynamic routing support for VRFs. |

Currently dynamic routing is supported for the following protocols:

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

 [IS-IS](https://docs.vyos.io/en/latest/configuration/protocols/isis.html#routing-isis)

 [OSPF](https://docs.vyos.io/en/latest/configuration/protocols/ospf.html#routing-ospf)

 [OSPFv3 (IPv6)](https://docs.vyos.io/en/latest/configuration/protocols/ospf.html#routing-ospfv3)

 [Static](https://docs.vyos.io/en/latest/configuration/protocols/static.html#routing-static)

The CLI configuration is same as mentioned in above articles. The only difference is, that each routing protocol used, must be prefixed with the *vrf name <name>* command.

**Example**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#example)

The following commands would be required to set options for a given dynamic routing protocol inside a given vrf:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|      | [BGP](https://docs.vyos.io/en/latest/configuration/protocols/bgp.html#routing-bgp): set vrf name <name> protocols bgp ...   |  | | --- | |  |   [IS-IS](https://docs.vyos.io/en/latest/configuration/protocols/isis.html#routing-isis): set vrf name <name> protocols isis ...   |  | | --- | |  |   [OSPF](https://docs.vyos.io/en/latest/configuration/protocols/ospf.html#routing-ospf): set vrf name <name> protocols ospf ...   |  | | --- | |  |   [OSPFv3 (IPv6)](https://docs.vyos.io/en/latest/configuration/protocols/ospf.html#routing-ospfv3): set vrf name <name> protocols ospfv3 ...   |  | | --- | |  |   [Static](https://docs.vyos.io/en/latest/configuration/protocols/static.html#routing-static): set vrf name <name> protocols static ...   |  | | --- | |  | |

**Operation**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#operation)

It is not sufficient to only configure a VRF but VRFs must be maintained, too. For VRF maintenance the following operational commands are in place.

|  |
| --- |
| **show vrf** |

Lists VRFs that have been created

|  |
| --- |
| vyos@vyos:~$ show vrf  VRF name state mac address flags interfaces -------- ----- ----------- ----- ---------- blue up 00:53:12:d8:74:24 noarp,master,up,lower\_up dum200,eth0.302  red up 00:53:de:02:df:aa noarp,master,up,lower\_up dum100,eth0.300,bond0.100,peth0 |

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

Command should probably be extended to list also the real interfaces assigned to this one VRF to get a better overview.  
 **show vrf <name>**

|  |
| --- |
| vyos@vyos:~$ show vrf name blue  VRF name state mac address flags interfaces -------- ----- ----------- ----- ---------- blue up 00:53:12:d8:74:24 noarp,master,up,lower\_up dum200,eth0.302 |

|  |
| --- |
| **show ip route vrf <name>** |

Display IPv4 routing table for VRF identified by *<name>*.

|  |
| --- |
| vyos@vyos:~$ show ip route vrf blue  Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR, f - OpenFabric, |

|  |
| --- |
| > - selected route, \* - FIB route, q - queued route, r - rejected route |

|  |
| --- |
| VRF blue:  K 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), 00:00:50 S>\* 172.16.0.0/16 [1/0] via 192.0.2.1, dum1, 00:00:02  C>\* 192.0.2.0/24 is directly connected, dum1, 00:00:06 |
| **show ipv6 route vrf <name>** |

Display IPv6 routing table for VRF identified by *<name>*.

|  |
| --- |
| vyos@vyos:~$ show ipv6 route vrf red  Codes: K - kernel route, C - connected, S - static, R - RIPng,  O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,  v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,  f - OpenFabric,  > - selected route, \* - FIB route, q - queued route, r - rejected route |

|  |
| --- |
| VRF red:  K ::/0 [255/8192] unreachable (ICMP unreachable), 00:43:20 C>\* 2001:db8::/64 is directly connected, dum1, 00:02:19 C>\* fe80::/64 is directly connected, dum1, 00:43:19  K>\* ff00::/8 [0/256] is directly connected, dum1, 00:43:19 |
| **ping <host> vrf <name>** |

The ping command is used to test whether a network host is reachable or not.

Ping uses ICMP protocol’s mandatory ECHO\_REQUEST datagram to elicit an ICMP   
ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams (pings) will have an IP and ICMP header, followed by “struct timeval” and an arbitrary number of pad bytes used to fill out the packet.

When doing fault isolation with ping, you should first run it on the local host, to verify that the local network interface is up and running. Then, continue with hosts and gateways further down the road towards your destination. Round-trip time and packet loss statistics are computed.

Duplicate packets are not included in the packet loss calculation, although the round-trip time of these packets is used in calculating the minimum/ average/maximum round-trip time numbers.

|  |
| --- |
| **Note** |
| Ping command can be interrupted at any given time using <Ctrl>+c . A brief statistic is shown   |  | | --- | |  |   afterwards. |

|  |
| --- |
|  |
| vyos@vyos:~$ ping 192.0.2.1 vrf red  PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.  64 bytes from 192.0.2.1: icmp\_seq=1 ttl=64 time=0.070 ms 64 bytes from 192.0.2.1: icmp\_seq=2 ttl=64 time=0.078 ms ^C --- 192.0.2.1 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 4ms rtt min/avg/max/mdev = 0.070/0.074/0.078/0.004 ms |

|  |
| --- |
| **traceroute vrf <name> [ipv4 | ipv6] <host>** |

Displays the route packets taken to a network host utilizing VRF instance identified by *<name>*.

When using the IPv4 or IPv6 option, displays the route packets taken to the given hosts IP address family. This option is useful when the host is specified as a hostname rather than an IP address.

|  |
| --- |
| **force vrf <name>** |

Join a given VRF. This will open a new subshell within the specified VRF.

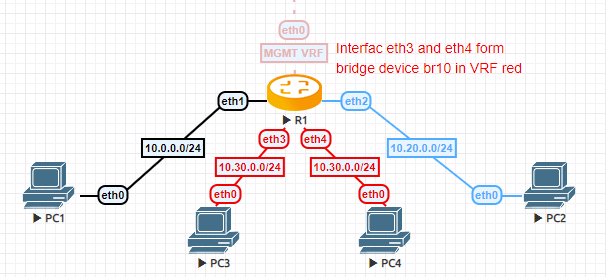
The prompt is adjusted to reflect this change in both config and op-mode.

|  |
| --- |
| vyos@vyos:~$ force vrf blue  vyos@vyos(vrf:blue):~$ |

**Example**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-example)

**VRF route leaking**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-route-leaking)

The following example topology was built using EVE-NG.



*VRF route leaking*[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#id9)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|    | PC1 is in the default VRF and acting as e.g. a “fileserver”   |  | | --- | |  |   PC2 is in VRF blue which is the development department   |  | | --- | |  |   PC3 and PC4 are connected to a bridge device on router R1 which is in VRF red .   |  | | --- | |  |  |  | | --- | |  | |

Say this is the HR department.

|  |  |  |
| --- | --- | --- |
|  | R1 is managed through an out-of-band network that resides in VRF mgmt   |  | | --- | |  | |

**Configuration**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-example-configuration)

|  |
| --- |
| set interfaces bridge br10 address '10.30.0.254/24'  set interfaces bridge br10 member interface eth3  set interfaces bridge br10 member interface eth4  set interfaces bridge br10 vrf 'red' |

|  |
| --- |
| set interfaces ethernet eth0 address 'dhcp'  set interfaces ethernet eth0 vrf 'mgmt'  set interfaces ethernet eth1 address '10.0.0.254/24' set interfaces ethernet eth2 address '10.20.0.254/24' set interfaces ethernet eth2 vrf 'blue' |

|  |
| --- |
| set protocols static route 10.20.0.0/24 interface eth2 vrf 'blue' set protocols static route 10.30.0.0/24 interface br10 vrf 'red' |

|  |
| --- |
| set service ssh disable-host-validation set service ssh vrf 'mgmt' |

|  |
| --- |
| set system name-server 'eth0' |

|  |
| --- |
| set vrf name blue protocols static route 10.0.0.0/24 interface eth1 vrf 'default' set vrf name blue table '3000'  set vrf name mgmt table '1000'  set vrf name red protocols static route 10.0.0.0/24 interface eth1 vrf 'default' set vrf name red table '2000' |

**VRF and NAT**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-and-nat)

**Configuration**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#id4)

|  |
| --- |
| set interfaces ethernet eth0 address '172.16.50.12/24' set interfaces ethernet eth0 vrf 'red' |

|  |
| --- |
| set interfaces ethernet eth1 address '192.168.130.100/24' set interfaces ethernet eth1 vrf 'blue' |

|  |
| --- |
| set nat destination rule 110 description 'NAT ssh- INSIDE'  set nat destination rule 110 destination port '2022'  set nat destination rule 110 inbound-interface 'eth0'  set nat destination rule 110 protocol 'tcp'  set nat destination rule 110 translation address '192.168.130.40' |

|  |
| --- |
| set nat source rule 100 outbound-interface 'eth0'  set nat source rule 100 protocol 'all'  set nat source rule 100 source address '192.168.130.0/24' set nat source rule 100 translation address 'masquerade' |

|  |
| --- |
| set service ssh vrf 'red' |

|  |
| --- |
| set vrf bind-to-all  set vrf name blue protocols static route 0.0.0.0/0 next-hop 172.16.50.1 vrf 'red' set vrf name blue protocols static route 172.16.50.0/24 interface eth0 vrf 'red' set vrf name blue table '1010' |

|  |
| --- |
| set vrf name red protocols static route 0.0.0.0/0 next-hop 172.16.50.1  set vrf name red protocols static route 192.168.130.0/24 interface eth1 vrf 'blue' |

|  |
| --- |
| set vrf name red table '2020' |

**Operation**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-example-operation)

After committing the configuration we can verify all leaked routes are installed, and try to ICMP ping PC1 from PC3.

|  |
| --- |
| PCS> ping 10.0.0.1 |

|  |
| --- |
| 84 bytes from 10.0.0.1 icmp\_seq=1 ttl=63 time=1.943 ms 84 bytes from 10.0.0.1 icmp\_seq=2 ttl=63 time=1.618 ms 84 bytes from 10.0.0.1 icmp\_seq=3 ttl=63 time=1.745 ms VPCS> show ip |

|  |
| --- |
| NAME : VPCS[1]  IP/MASK : 10.30.0.1/24  GATEWAY : 10.30.0.254  DNS :  MAC : 00:50:79:66:68:0f |

VRF default routing table[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-default-routing-table)

|  |
| --- |
| vyos@R1:~$ show ip route  Codes: K - kernel route, C - connected, S - static, R - RIP,  O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,  F - PBR, f - OpenFabric,  > - selected route, \* - FIB route, q - queued, r - rejected, b - backup |

|  |
| --- |
| C>\* 10.0.0.0/24 is directly connected, eth1, 00:07:44  S>\* 10.20.0.0/24 [1/0] is directly connected, eth2 (vrf blue), weight 1, 00:07:38 S>\* 10.30.0.0/24 [1/0] is directly connected, br10 (vrf red), weight 1, 00:07:38 VRF red routing table[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-red-routing-table) |

|  |
| --- |
| vyos@R1:~$ show ip route vrf red  Codes: K - kernel route, C - connected, S - static, R - RIP,  O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,  F - PBR, f - OpenFabric,  > - selected route, \* - FIB route, q - queued, r - rejected, b - backup |

|  |
| --- |
| VRF red:  K>\* 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), 00:07:57  S>\* 10.0.0.0/24 [1/0] is directly connected, eth1 (vrf default), weight 1, 00:07:40 C>\* 10.30.0.0/24 is directly connected, br10, 00:07:54 |

VRF blue routing table[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#vrf-blue-routing-table)

|  |
| --- |
| vyos@R1:~$ show ip route vrf blue  Codes: K - kernel route, C - connected, S - static, R - RIP,  O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,  T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,  F - PBR, f - OpenFabric,  > - selected route, \* - FIB route, q - queued, r - rejected, b - backup |

|  |
| --- |
| VRF blue:  K>\* 0.0.0.0/0 [255/8192] unreachable (ICMP unreachable), 00:08:00 |

|  |
| --- |
| S>\* 10.0.0.0/24 [1/0] is directly connected, eth1 (vrf default), weight 1, 00:07:44 C>\* 10.20.0.0/24 is directly connected, eth2, 00:07:53 |

**L3VPN VRFs**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#l3vpn-vrfs)

L3VPN VRFs bgpd supports for IPv4 RFC 4364 and IPv6 RFC 4659. L3VPN routes, and their associated VRF MPLS labels, can be distributed to VPN SAFI neighbors in the default, i.e., non VRF, BGP instance. VRF MPLS labels are reached using core MPLS labels which are distributed using LDP or BGP labeled unicast. bgpd also supports inter-VRF route leaking.

**VRF Route Leaking**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#l3vpn-vrf-route-leaking)

BGP routes may be leaked (i.e. copied) between a unicast VRF RIB and the VPN SAFI RIB of the default VRF for use in MPLS-based L3VPNs. Unicast routes may also be leaked between any VRFs (including the unicast RIB of the default BGP instance). A shortcut syntax is also available for specifying leaking from one VRF to another VRF using the default instance’s VPN RIB as the intemediary . A common application of the VRF-VRF feature is to connect a customer’s private routing domain to a provider’s VPN service. Leaking is configured from the point of view of an individual VRF: import refers to routes leaked from VPN to a unicast VRF, whereas export refers to routes leaked from a unicast VRF to VPN.

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

|  |
| --- |
| Routes exported from a unicast VRF to the VPN RIB must be augmented by two parameters: |

|  |
| --- |
| an RD / RTLIST |

|  |
| --- |
| Configuration for these exported routes must, at a minimum, specify these two parameters. |

**Configuration**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#l3vpn-vrf-example-configuration)

Configuration of route leaking between a unicast VRF RIB and the VPN SAFI RIB of the default VRF is accomplished via commands in the context of a VRF address-family.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> rd vpn export <asn:nn|address:nn>** |

Specifies the route distinguisher to be added to a route exported from the current unicast VRF to VPN.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> route-target vpn <import|export|both> [RTLIST]** |

Specifies the route-target list to be attached to a route (export) or the route-target list to match against (import) when exporting/importing between the current unicast VRF and VPN.The RTLIST is a space-separated list of route-targets, which are BGP extended community values as described in Extended Communities Attribute.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> label vpn export <0-1048575|auto>** |

Enables an MPLS label to be attached to a route exported from the current unicast VRF to VPN. If the value specified is auto, the label value is automatically assigned from a pool maintained.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> label vpn allocation-mode per-nexthop** |

Select how labels are allocated in the given VRF. By default, the per-vrf mode is selected, and one label is used for all prefixes from the VRF. The per-nexthop will use a unique label for all prefixes that are reachable via the same nexthop.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> route-map vpn <import|export> [route-map <name>]** |

Specifies an optional route-map to be applied to routes imported or exported between the current unicast VRF and VPN.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> <import|export> vpn** |

Enables import or export of routes between the current unicast VRF and VPN.

|  |
| --- |
| **set vrf name <name> protocols bgp address-family <ipv4-unicast|ipv6-unicast> import vrf <name>** |

Shortcut syntax for specifying automatic leaking from vrf VRFNAME to the current VRF using the VPN RIB as intermediary. The RD and RT are auto derived and should not be specified explicitly for either the source or destination VRF’s.

|  |
| --- |
| **set vrf name <name> protocols bgp interface <interface> mpls forwarding** |

It is possible to permit BGP install VPN prefixes without transport labels. This configuration will install VPN prefixes originated from an e-bgp session, and with the next-hop directly connected.

**Operation**[](https://docs.vyos.io/en/latest/configuration/vrf/index.html#l3vpn-vrf-example-operation)

It is not sufficient to only configure a L3VPN VRFs but L3VPN VRFs must be maintained, too.For L3VPN VRF maintenance the following operational commands are in place.

|  |
| --- |
| **show bgp <ipv4|ipv6> vpn** |

Print active IPV4 or IPV6 routes advertised via the VPN SAFI.

|  |
| --- |
| BGP table version is 2, local router ID is 10.0.1.1, vrf id 0  Default local pref 100, local AS 65001  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  Route Distinguisher: 10.50.50.1:1011  \*>i10.50.50.0/24 10.0.0.7 0 100 0 i  UN=10.0.0.7 EC{65035:1011} label=80 type=bgp, subtype=0 Route Distinguisher: 10.60.60.1:1011  \*>i10.60.60.0/24 10.0.0.10 0 100 0 i  UN=10.0.0.10 EC{65035:1011} label=80 type=bgp, subtype=0 |
| **show bgp <ipv4|ipv6> vpn summary** |

Print a summary of neighbor connections for the specified AFI/SAFI combination.

|  |
| --- |
| BGP router identifier 10.0.1.1, local AS number 65001 vrf-id 0 BGP table version 0  RIB entries 9, using 1728 bytes of memory  Peers 4, using 85 KiB of memory  Peer groups 1, using 64 bytes of memory |

|  |
| --- |
| Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd PfxSnt 10.0.0.7 4 65001 2860 2870 0 0 0 1d23h34m 2 10 |