VyOS

an open source network operating system for routers

Why VyOS?

- A carrier grade platform available to everyone
- Open and community-driven nature of development
- Works on a wide range of off the shelf hardware and virtual platforms
- A router you can bring to any network — from a small office to a datacenter rack
- Time proven — continued development of a now abandoned Vyatta Core system

Quick facts

- More than 10k deployments of all sizes worldwide
- Focus on enterprise and service provider networks but not limited to them
- Command line interface in the style of JunOS
- Completely free and open source, community-driven development
- Runs on x86 and all major hypervisors (KVM, Xen, VMware, Hyper-V), experimental builds for ARM
- Available on AWS (GCP and Azure are on the roadmap)
- Commercial support and professional services are available

Where VyOS comes from?

VyOS is a fork of now defunct Vyatta system.

- 2006: Vyatta Community (later Vyatta Core) first release
- 2011: Ubiquiti Networks makes a fork of Vyatta under EdgeOS name and starts the EdgeMax product line with it
- 2012: Open source Vyatta development stalls, the original company shifts the focus to a proprietary version, later gets acquired by Brocade who renames it to Brocade Vyatta vRouter 5400
- 2013: Any hope that Vyatta Core development will resume is gone, a group of users starts a fork under VyOS name

Development philosophy

- Open to contributions of all kinds (patches, design proposals, feature requests, documentation, testing, anything)
- Maintainers don’t keep anything to themselves: everything required to build it from scratch is public
- Documented APIs for integrating new features
- Security and stability is a priority
- One-step image build process, users can build custom images for their needs
- Not controlled by any corporate entity (i.e. won’t just disappear one day at their will)
- The source code of the main system is under the GNU GPL, no contributor agreements, therefore no legal possibility of a closed-source fork

Feature checklist

- Network: 802.1q VLANs, 802.1ad QinQ, 802.3ad LACP and other bonding types, bridging, mirroring and redirection
- Routing protocols: BGP (IPv4 and IPv6), OSPFv2, RIP, RIPng, IGMP proxy
- Firewall: IPv4 and IPv6, stateful filtering, zone-based policies, address/port/network groups for IPv4
- NAT: Source NAT, port-address translation, one-to-many and many-to-many translations(CGNAT)
- VPN: Site-to-site IPsec, VTI, DMVPN, PPTP and L2TP servers, OpenVPN (client-server and s2s)
- Tunnels: GRE, IPIP, SIT, IP6IP, umanaged L2TPv3, VXLAN
- Redundancy: VRRP for IPv4 (IPv6 in the upcoming version), conntrack sync, WAN failover
- <Your feature request here>

Reliability and robustness

- Versioned upgrades: if something goes wrong, you can reboot to previous version
- Versioned configuration: you can view previous revisions and diffs and rollback
- Stateful configuration system: no actual changes are made until you issue a commit command
- Multi-user friendly CLI: changes are staged into user’s sessions, users are notified of one another’s changes

Taste of the CLI

```
vyos@router:~$ show interfaces tunnel
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                               S/L  Description
---------        ----------                               ---  -----------
tun0             10.10.10.1/30                            u/u
vyos@router:~$ configure
vyos@router# delete interfaces tunnel tun0 address 10.10.10.1/30
vyos@router# set interfaces tunnel tun0 address 10.10.11.1/30
vyos@router# show interfaces tunnel
 tunnel tun0 {
-    address 10.10.10.1/30
+    address 10.10.11.1/30
     encapsulation gre
     ip {
         ospf {
>            bandwidth 1000
         }
     }
     local-ip 192.168.99.1
     remote-ip 192.168.99.2
 }
vyos@router# commit
vyos@router# run show interfaces tunnel
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                               S/L  Description
---------        ----------                               ---  -----------
tun0             10.10.11.1/30                            u/u
```

operational mode command

entering configuration mode

configuration diff

only now changes are active, one could discard them instead

operational mode command ran from configuration mode

Configuration system

- All configuration data is kept in /config directory, easy to move between systems.
- Most of the configuration in one text file: /config/config.boot
- Built-in configuration versioning and backup to TFTP/SCP/etc., no need for RANCID-like tools
- Post-config, and pre-commit/post-commit hooks: add your own fixups, custom consistency checks or backup scripts and so on
- Stateful configuration: no actual changes are made until you hit commit
- Multi-user friendly: everyone gets their own session, you can't accidentally commit someone else's changes, you are notified when someone commits
- Configuration is easily observable with show commands (“show”, “show interfaces”, “show interfaces ethernet”, ...)
- You can get the set commands for current configuration (“run show configuration commands”) and apply on another router
- Merging configuration file snippets into current configuration is supported too
- You can view diffs between configuration revisions and rollback to previous versions (requires reboot, we are looking for solutions)
- If changes can be dangerous, use “commit-confirm” command that reboots into previous revision unless you confirm the changes.

Safe and easy upgrades

```
vyos@vyos$ show system image
The system currently has the following image(s) installed:
   1: VyOS-1.1.6 (default boot) (running image)
vyos@vyos:~$ add system image http://mirror.vyos.net/iso/release/1.1.7/vyos-1.1.7-amd64.iso
Trying to fetch ISO file from http://mirror.vyos.net/iso/release/1.1.7/vyos-1.1.7-amd64.iso
  [SNIP]
Done.
vyos@vyos$ show system image
The system currently has the following image(s) installed:
   1: VyOS-1.1.7 (default boot)
   2: VyOS-1.1.6 (running image)
```

No changes made to current installation, images are kept separately.
If something goes wrong, you can boot to another version.
If config syntax changes between versions, migration scripts will convert the config file to new version.

Downside: upgrade requires reboot.
We are looking for solutions (Nix package manager perhaps?)

Extensive support for physical and virtual usage

- Small x86 boxes (Alix, Netgate etc.), as long as it has at least 4GB flash drive
- Big x86 servers
- Virtualization platforms: KVM, Xen, VMWare ESXi, Hyper-V. Paravirtual drivers (virtio, vmxnet3, ...) and open-vm-tools for VMware are included in the image, not limited to emulated hardware.
- Cloud platforms. Currently there's official AMI on AWS Marketplace and an image on Ravello, Google Cloud is coming soon. You can make your own and send us patches/build scripts!
- Experimental builds for ARM

We believe that routers should adapt to people's needs rather than other way around and we call it “a router you can bring to any network”, from small office to a datacenter rack.

However, no support for small routers like WRT54G etc., there's OpenWRT/DD-WRT and they doing great job there

Extensible system

- Scripts that generate actual iptables/quagga/etc. configs read the system config through a library in a unified way
- Config read API and command definition syntax is documented
- Package skeleton is provided
- Additional packages with new commands can be installed on a live system for testing, or added to a custom-built image
- A few complete features made by our users already have been merged into the image, not every feature is initiated by the maintainers
- Complete build toolchain is available to everyone, contributors can have exact same setup that maintainers use for building release images

Case study #1: AVyS Telecom S.A.

- A voice over IP operator (lots of small packets)
- Bandwidth 1GBps
- Cisco 3925 at the primary site was replaced with two instances of VyOS running on VMware in VRRP pair for the BGP full feed, another instance for VPNs and internal network. Saved a few rack units and quite some power.
- Secondary site was built with virtual VyOS in mind from the start, two routers for BGP full feed in VRRP pair as well, VPN router. A couple of routers was added specially for storage replication traffic, over a GRE/IPsec tunnel (about 200MBps encrypted traffic).
- At the offices, a DSL modem was connected to a “router on a stick” VyOS VM through a VLAN, no need for a router with DSL ports
- Risky changes are routinely tested on disposable test VMs before applying them to production routers.
- Training new network admins unfamiliar with VyOS took just a couple of months, they could perform routine tasks after the first week

Case study #2: Bluephone S.L.

Mobile Virtual Network Operator under FlexiMovil brand
Greenfield project:

- No physical routers
- Two VyOS instances in VRRP pair on VMWare vSphere for the BGP full feed
- Two VyOS instances in VRRP interconnected with MVN Enabler infrastructure(Redundant pair of GGSNs) for mobile subscriber traffic
- Lots of small instances for testing and development environments

IPsec performance test

User-contributed performance test. The setup:

- Hypervisor: VMware
- IPsec parameters: AES256 cipher, SHA-1 hash

Results:

- On a 2.0GHz Xeon CPU: 450MBps
- On a 2.6GHz Xeon CPU: 635MBps

See https://whiskeyalpharomeo.com/2016/09/12/vyos-crypto-throughput-initial-results for details.

The future

- VyOS 1.2.0: uses Debian Jessie codebase, multiple new features. Currently in beta, testers are welcome!
- VyOS 2.0: planned clean rewrite to fix long-lasting design issues in the aging codebase and greatly further reliability and robustness of the system, such as configuration rollbacks and upgrades without reboot. Everyone is welcome to join the design discussions and development!

Join the VyOS project!

Web: https://vyos.io

Development: https://phabricator.vyos.net

Forum: http://forum.vyos.net

IRC: #vyos on Freenode IRC network

RocketChat: https://chat.vyos.io

Social Media

Twitter: https://twitter.com/vyos_dev

Facebook: https://www.facebook.com/vyosofficial