

Debian + Proxmox + VYOS + NAT + IPSec + IPTables + Cheap Servers = Fun (?!)

Version 1.0

Saturday, 12 March 2016 (this is not my day job)

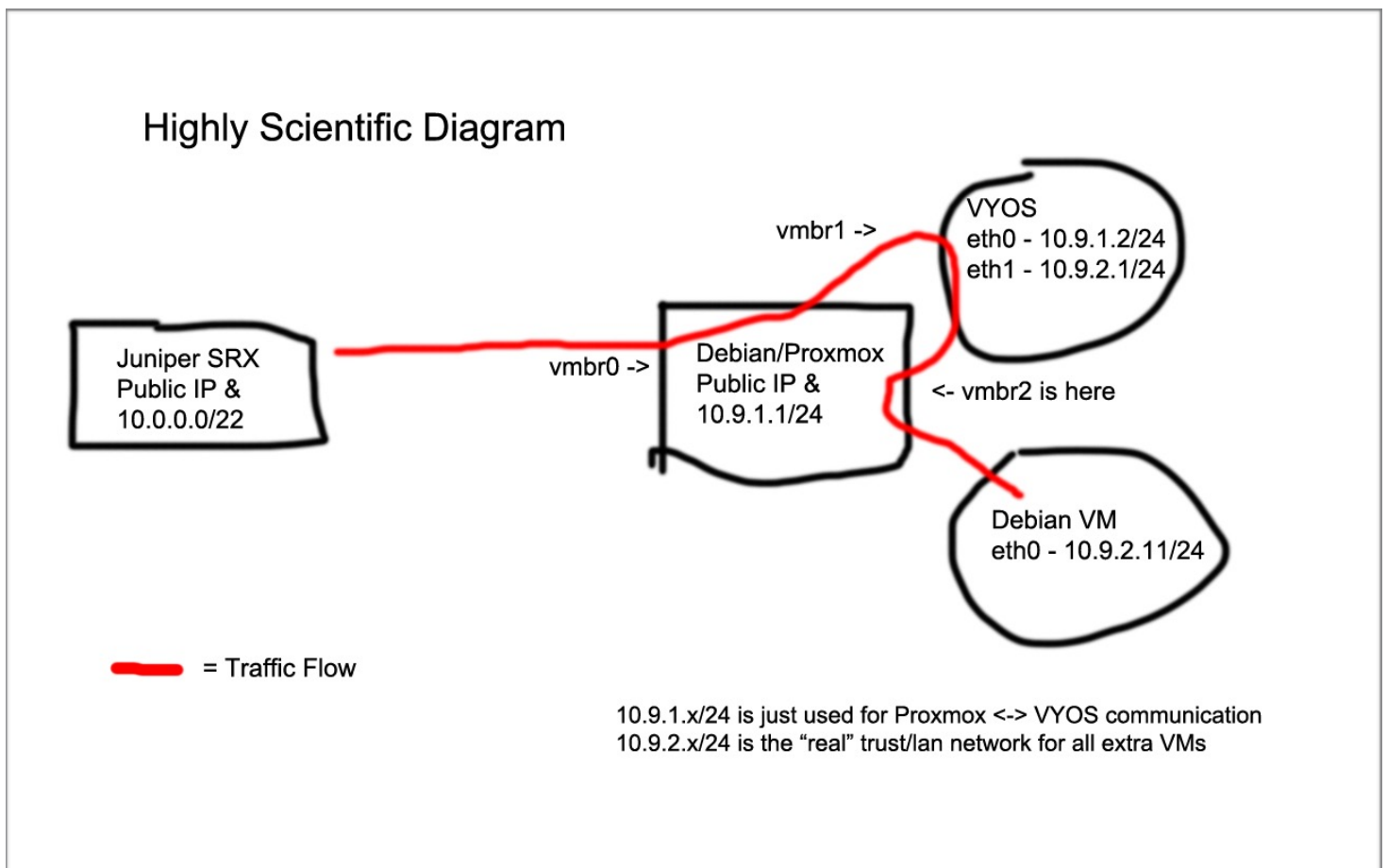
Michael Dale

michael@dalegroup.net

This is basically a rambling blog/guide on how to setup the above.

This isn't designed to be used in a production environment.

I might update the doc if there are suggestions/bugs etc.



Background

Recently I picked up a very cheap dedicated server in the US from - <http://delimiter.com>

The original reason for purchasing it was the price, and price alone. I already have a bunch of servers so I didn't need another, but I was keen to try out something so cheap.

I ended up with:

- 2x Xeon E5420 (so quad core CPUs at 2.5GHz, for a total of 8 cores).
- 2x 500GB HDDs (the ones I ended up with are WD Blue drives, basically pretty awful, but for the cost completely expected).
- 16GB Ram (DDR2)
- 1GB port with 10TB bandwidth
- 1 IPv4 Address
- This is all on a blade HP BL260c G5. Far from the newest but you do get iLO

The price is normally \$US30/month but if you pay for the year it ends up being only \$US200. So less than \$AU25/month for a dedicated server! Crazy price.

Anyway so I was thinking about what I wanted to use it for, and wasn't really sure.

I normally use ESXi so I installed that first. Unfortunately the BL260c doesn't have a real raid controller so it would have ended up being to separate 500GB drives. Also ESXi is known for running badly without a battery backed up raid controller.

I decided to scrap that idea. Although I'm not planning to do anything critical on this box I wasn't happy running without at least Raid 1.

I normally use Debian on our servers at work as this has worked well for many years and proved very easy to maintain and upgrade when required.

You can easily re-install OSs on the box via the Delimiter control panel, and complete the install via iLO. Although it seems most images they have are setup to fully auto install, which is nice.

Unfortunately the Debian Jessie image they have seemed to have issues for me. It could never complete the install due to network issues downloading from various mirrors.

I ended up installing Debian 7 and then manually upgrading to 8. The Debian 7 image template gave me the option to auto setup a software Raid 1, perfect, so after install I didn't need to do much.

Now I'm stuck with this overly powerful Debian box with nothing to do! So I decided it would be fun to play with Proxmox, which has been on my list for a while.

Proxmox Setup

My understanding of Proxmox is that it is basically KVM with a web front-end for managing it. Allowing you to easily manage/install VMs. Great! There is a guide online explaining how to install Proxmox on Debian https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Jessie

I ignored this almost completely and just installed the package via apt-get install, seemed to work fine.

I think the steps I took were:

- `echo "deb http://download.proxmox.com/debian jessie pve-no-subscription" > /etc/apt/sources.list.d/pve-install-repo.list`
- `wget -O- "http://download.proxmox.com/debian/key.asc" | apt-key add -`
- `apt-get update && apt-get dist-upgrade`
- `apt-get install proxmox-ve ntp ssh postfix ksm-control-daemon open-iscsi systemd-sysv`

Once installed the networking side of things was a bit confusing for me as I am used to the way ESXi works.

It ended up being easiest to setup all the networking directly in the `/etc/network/interfaces` file.

Here is what I've ended up with, I will explain below (sorry for the small font, easier to fit on one line).

Once installed you can access the Web UI from: `https://youripaddress:8006`

You use your Debian root username & password.

Debian/Proxmox Host /etc/network/interfaces file.

```
auto lo
iface lo inet loopback

allow-hotplug eth0

iface eth0 inet manual

iface eth1 inet manual

auto vmbr0
iface vmbr0 inet static
    address xxx.xxx.xxx.xxx
    netmask 255.255.255.192
    gateway xxx.xxx.xxx.xxx
    broadcast xxx.xxx.xxx.xxx
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    network xxx.xxx.xxx.xxx
    dns-nameservers 8.8.8.8
    dns-search example.local
# dns-* options are implemented by the resolvconf package, if installed

auto vmbr1
iface vmbr1 inet static
    address 10.9.1.1
    netmask 255.255.255.0
    bridge_ports none
    bridge_stp off
    bridge_fd 0

auto vmbr2
iface vmbr2 inet manual
    bridge_ports none
    bridge_stp off
    bridge_fd 0

post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -o vmbr0 -j MASQUERADE
post-up iptables -t nat -A PREROUTING -i vmbr0 -p tcp -m tcp --dport 11122 -j DNAT --to-destination 10.9.1.2:22
post-up iptables -t nat -A PREROUTING -i vmbr0 -p tcp -m tcp --dport 11123 -j DNAT --to-destination 10.9.2.11:22
post-up iptables -t nat -A PREROUTING -i vmbr0 -p udp -m udp --dport 500 -j DNAT --to-destination 10.9.1.2:500
post-up iptables -t nat -A PREROUTING -i vmbr0 -p udp -m udp --dport 4500 -j DNAT --to-destination 10.9.1.2:4500
post-up iptables -t nat -A PREROUTING -i vmbr0 -p 50 -j DNAT --to-destination 10.9.1.2
post-up iptables -t nat -A PREROUTING -i vmbr0 -p 51 -j DNAT --to-destination 10.9.1.2

post-up route add -net 10.9.0.0 netmask 255.255.0.0 gw 10.9.1.2

post-down iptables -t nat -D POSTROUTING -o vmbr0 -j MASQUERADE
```

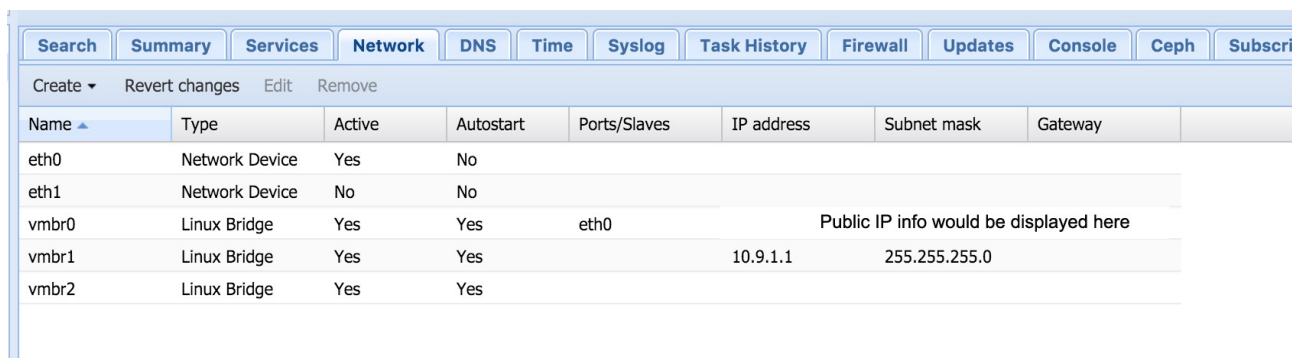
So the physical server has 2 NICs, eth0 and eth1. Only eth0 is actually used. eth1 is not connected.

By default eth0 will have all your public IP address info. We need to move this over to vmbr0 (basically a virtual interface) and then link it back with eth0. This is what you can see in after "iface vmbr0 inet static"

Next we create vmbr1, this going to be the "LAN" side of the Proxmox/Debian box. This is the interface that VYOS will connect to (for VYOS this is basically the "WAN" side). You can use any internal IP range/scheme you wish. For this I have used 10.9.1.1/24.

Finally we have vmbr2, as you can see we do not have an IP address on this interface. We are treating this interface as the VYOS LAN side and do not want the Proxmox box talking directly on this interface/network, it needs to go through vmbr1.

In the Proxmox UI it looks like this:



The screenshot shows the Proxmox Network configuration page. At the top, there are tabs for Search, Summary, Services, Network (selected), DNS, Time, Syslog, Task History, Firewall, Updates, Console, Ceph, and Subscri. Below the tabs, there are buttons for Create, Revert changes, Edit, and Remove. The main content is a table with the following columns: Name, Type, Active, Autostart, Ports/Slaves, IP address, Subnet mask, and Gateway.

Name	Type	Active	Autostart	Ports/Slaves	IP address	Subnet mask	Gateway
eth0	Network Device	Yes	No				
eth1	Network Device	No	No				
vmbr0	Linux Bridge	Yes	Yes	eth0			Public IP info would be displayed here
vmbr1	Linux Bridge	Yes	Yes		10.9.1.1	255.255.255.0	
vmbr2	Linux Bridge	Yes	Yes				

You may need to reboot and/or restart the networking after making changes to the interfaces file. If you make a mistake you can login via iLO.

After all the interface configuration comes some of the complex/fun part of getting this crazy setup to work.

As we only have a single public IP address we need to NAT certain connections into the VYOS box.

The first 2 lines (the ip_forward and MASQUERADE bit) do all the basics. This will allow VYOS to connect externally (you should now be able to ping 8.8.8.8 from the VYOS VM).

The next lines NAT through SSH/IPSec to the VYOS box. We also have another SSH nat through to a VM sitting behind the VYOS box (this is on the VYOS LAN side, 10.9.2.11)

This allows me to create an IPSec site-to-site VPN with my firewall at home (Juniper SRX210). Meaning that anything on the VYOS LAN side (10.9.2.x) I can connect to directly from home.

Port Forwards

The following port forwards:

```
post-up iptables -t nat -A PREROUTING -i vbr0 -p tcp -m tcp --dport 11122 -j DNAT --to-destination 10.9.1.2:22
post-up iptables -t nat -A PREROUTING -i vbr0 -p udp -m udp --dport 500 -j DNAT --to-destination 10.9.1.2:500
post-up iptables -t nat -A PREROUTING -i vbr0 -p udp -m udp --dport 4500 -j DNAT --to-destination 10.9.1.2:4500
post-up iptables -t nat -A PREROUTING -i vbr0 -p 50 -j DNAT --to-destination 10.9.1.2
post-up iptables -t nat -A PREROUTING -i vbr0 -p 51 -j DNAT --to-destination 10.9.1.2
```

Allows IPsec and SSH to pass through. As SSH is also enabled on the Proxmox box I've forwarded 11122 to 22 on the VYOS box.

So I can now connect directly to the VYOS box doing `ssh vyos@public_ip_address 11122`

The other port forwards are for IPsec (50, 51, 500 and 4500). This allows you to setup a standard site-to-site IPsec VPN to your public IP address.

As the VYOS box does not have a public IP address the ike ID it sends will be its internal address and not your public IP address.

On JunOS there is a command to ignore the incorrect ike ID. Which we will need to use, more in the JunOS section.

Setting up VYOS

VYOS is a community fork of Vyatta. EdgeOS (from ubiquiti) is also based on this. Basically it is like a mini Juniper JunOS firewall/router in a VM. Not as powerful and many of the commands are not laid out as nicely, overall I prefer JunOS, but VYOS is free and easy to get, which is a massive plus.

EdgeOS is actually proving to be an awesome fork of it. If I could I would run EdgeOS over VYOS.

But I can't, so here we are.

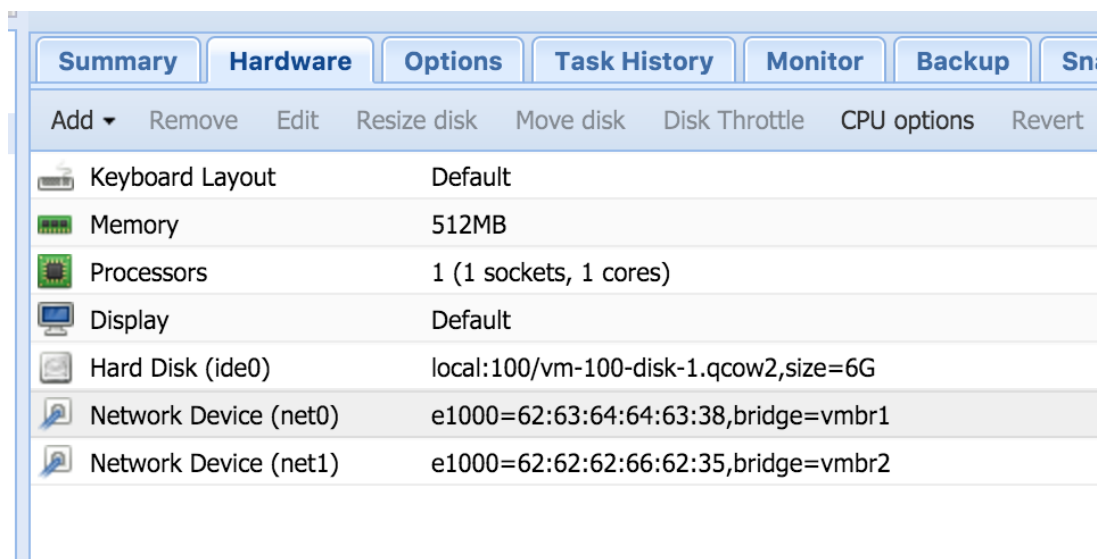
VYOS is completely fine and works pretty well. You could probably replace VYOS with something like PfSense, but I do like my command line firewalls.

Actually you could probably not have a firewall VM at all and just use iptables in Proxmox, but where is the fun in that?

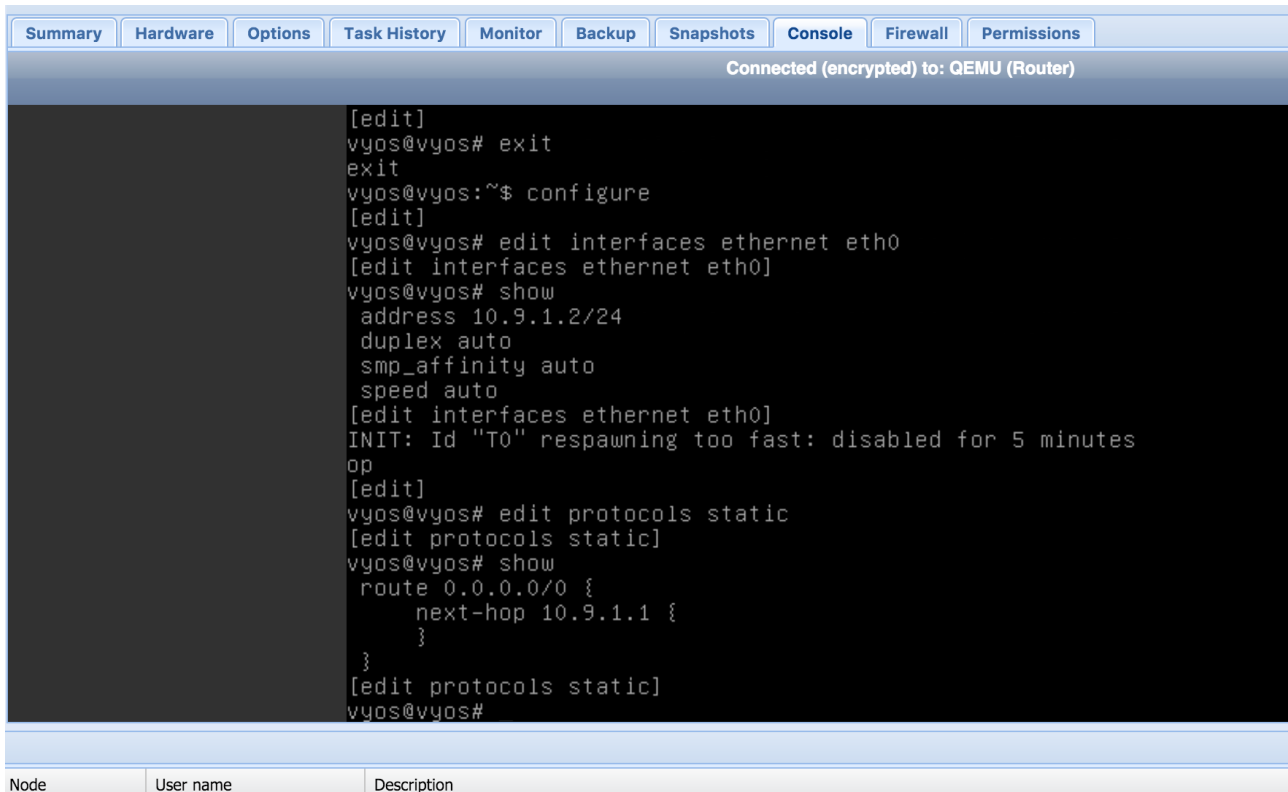
Lets setup VYOS!

- Download the AMD64 iOS of VYOS, at the time version 1.1.7 was the latest: <http://vyos.nervex.net/iso/release/1.1.7/vyos-1.1.7-amd64.iso>
- Copy the ISO file to the Debian/Proxmox Host to the folder: `/var/lib/vz/template/iso`
- Now you can create the VM. You should be able to select the IOS for the CD, the OS type is just the latest Linux option, 6GB for HDD is more than enough, 512MB should also be plenty for the ram. Leave the HDD options default.
- When you add the network ensure you select bridge mode and select the interface vmb1, don't change any other settings like VLAN or Firewall etc. I used the E1000 driver but the others will be better, I just didn't test them in VYOS.
- VMBR1 will then map to NET0 and then ETH0 within the VYOS VM. This will be seen as the "WAN" interface.
- You then need to create another network card, this can be added later. This interface will be VMBR2, again don't change the settings on VLAN etc. This will become the LAN interface on VYOS.
- Remember that we didn't add any IP address on VMBR2 on the host box, this is because all traffic will flow through VYOS VM.

This is what it looks like in Proxmox.



Here is what the “WAN” side looks like from VYOS, including the default route.

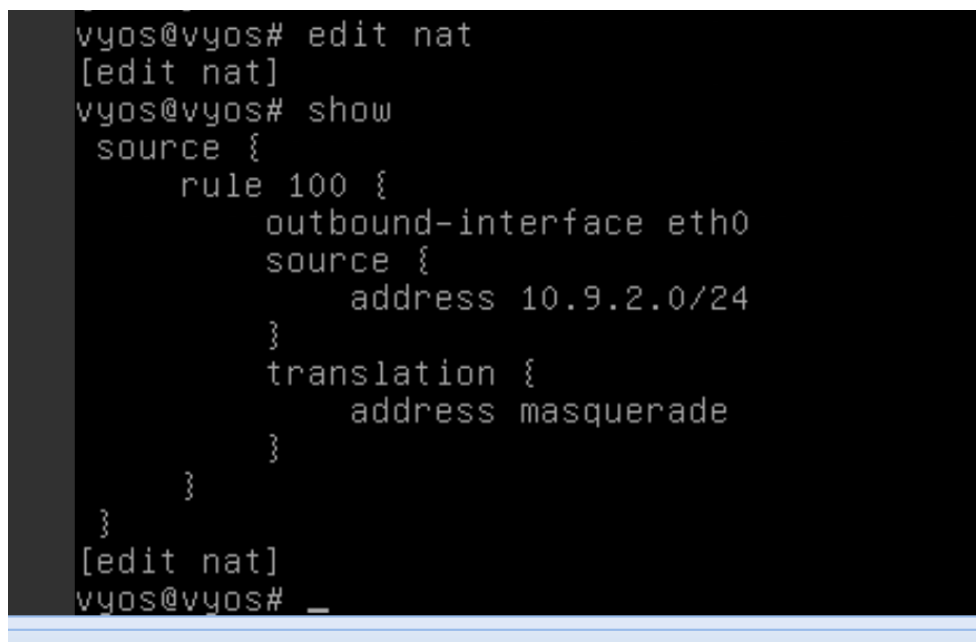


```
Summary Hardware Options Task History Monitor Backup Snapshots Console Firewall Permissions
Connected (encrypted) to: QEMU (Router)

[edit]
vyos@vyos# exit
exit
vyos@vyos:~$ configure
[edit]
vyos@vyos# edit interfaces ethernet eth0
[edit interfaces ethernet eth0]
vyos@vyos# show
address 10.9.1.2/24
duplex auto
smp_affinity auto
speed auto
[edit interfaces ethernet eth0]
INIT: Id "TO" respawning too fast: disabled for 5 minutes
op
[edit]
vyos@vyos# edit protocols static
[edit protocols static]
vyos@vyos# show
route 0.0.0.0/0 {
    next-hop 10.9.1.1 {
    }
}
[edit protocols static]
vyos@vyos#
```

Node	User name	Description
------	-----------	-------------

You also need to ensure that you have NAT setup on the VYOS box.



```
vyos@vyos# edit nat
[edit nat]
vyos@vyos# show
source {
    rule 100 {
        outbound-interface eth0
        source {
            address 10.9.2.0/24
        }
        translation {
            address masquerade
        }
    }
}
[edit nat]
vyos@vyos# _
```

Once NAT is setup you should be able to ping out from the VYOS box.

Now there are just 3-4 last steps on the VYOS box.

- LAN Network Interface
- DHCP for new virtual LAN network
- IPsec Tunnel
- Security Policies (not going to talk about that for now)

So just as you have setup eth0 you now need to create eth1.

eth1 can be any subnet you want. For this I have used 10.9.2.1/24, this means for any VMs behind the VYOS box, 10.9.2.1 will become the default gateway.

This step is easy, see the config below for more info.

Remember eth1 maps to vubr2 on the Proxmox side. Any new VM you create that you want on the VYOS LAN/Trust network side you ensure that you add the network using vubr2.

Also time to setup DHCP. This just makes it easy when you're setting up new VMs, you don't need to use DHCP if you want to static set all your VMs.

SECURITY!!!!

VYOS is just forwarding everything, no firewall etc has been setup. As you are natting from Proxmox it probably isn't a world ending issue, but please be aware of it.

The IPsec side of things isn't too difficult, but best just to modify for your environment. I am using IPsec Main Mode Auth as I have a static IP address at home too. But you may need to use aggressive mode, or even something like OpenVPN if you're not using a Juniper at home.

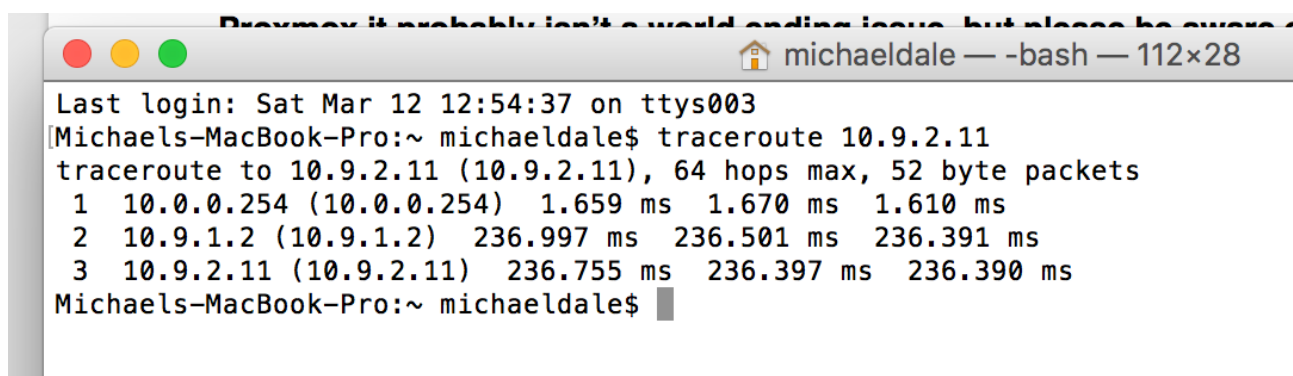
Check the config below.

At home my network is on 10.0.0.0/22.

The IPsec Tunnel allows 10.0.0.0/22 to talk directly to the 10.9.2.0/24 (Virtual/Trust LAN side) on VYOS directly.

It looks like this from my laptop. 10.0.0.254 is my home Juniper device. 10.9.1.2 is my VYOS "WAN" side and 10.9.2.11 is my new VM behind the VYOS box.

Yes latency from Australia is __bad__



```
Proxmox it probably isn't a world ending issue, but please be aware of it
michaeldale — -bash — 112x28
Last login: Sat Mar 12 12:54:37 on ttys003
Michaels-MacBook-Pro:~ michaeldale$ traceroute 10.9.2.11
traceroute to 10.9.2.11 (10.9.2.11), 64 hops max, 52 byte packets
 1 10.0.0.254 (10.0.0.254)  1.659 ms  1.670 ms  1.610 ms
 2 10.9.1.2 (10.9.1.2)  236.997 ms  236.501 ms  236.391 ms
 3 10.9.2.11 (10.9.2.11)  236.755 ms  236.397 ms  236.390 ms
Michaels-MacBook-Pro:~ michaeldale$
```

VYOS Configuration

```
vyos@vyos# show | no-more
interfaces {
  ethernet eth0 {
    address 10.9.1.2/24
    duplex auto
    smp_affinity auto
    speed auto
  }
  ethernet eth1 {
    address 10.9.2.1/24
    duplex auto
    smp_affinity auto
    speed auto
  }
  loopback lo {
  }
}
nat {
  source {
    rule 100 {
      outbound-interface eth0
      source {
        address 10.9.2.0/24
      }
      translation {
        address masquerade
      }
    }
  }
}
protocols {
  static {
    route 0.0.0.0/0 {
      next-hop 10.9.1.1 {
      }
    }
  }
}
service {
  dhcp-server {
    disabled false
    shared-network-name trust {
      subnet 10.9.2.0/24 {
        default-router 10.9.2.1
        dns-server 8.8.8.8
        start 10.9.2.100 {
          stop 10.9.2.200
        }
      }
    }
  }
  dns {
  }
  ssh {
    port 22
  }
}
system {
  config-management {
    commit-revisions 20
  }
  console {
    device ttyS0 {
      speed 9600
    }
  }
  host-name vyos
  login {
    user vyos {
      authentication {
        encrypted-password xxxxx
        plaintext-password ""
      }
      level admin
    }
  }
  name-server 8.8.8.8
  ntp {
    server 0.pool.ntp.org {
    }
    server 1.pool.ntp.org {
    }
    server 2.pool.ntp.org {
    }
  }
  package {
```

```

    auto-sync 1
    repository community {
        components main
        distribution helium
        password ""
        url http://packages.vyos.net/vyos
        username ""
    }
}
syslog {
    global {
        facility all {
            level notice
        }
        facility protocols {
            level debug
        }
    }
}
time-zone UTC
}
vpn {
    ipsec {
        esp-group esp-co {
            compression disable
            lifetime 3600
            mode tunnel
            pfs dh-group2
            proposal 1 {
                encryption aes128
                hash sha1
            }
        }
        ike-group co {
            lifetime 28800
            proposal 1 {
                dh-group 2
                encryption aes128
                hash sha1
            }
        }
        ipsec-interfaces {
            interface eth0
        }
        site-to-site {
            peer xxx.xxx.xxx.xx {
                authentication {
                    mode pre-shared-secret
                    pre-shared-secret xxxxx
                }
                connection-type initiate
                default-esp-group esp-co
                ike-group co
                local-address 10.9.1.2
                tunnel 1 {
                    local {
                        prefix 10.9.0.0/16
                    }
                    protocol all
                    remote {
                        prefix 10.0.0.0/22
                    }
                }
            }
        }
    }
}
}
}

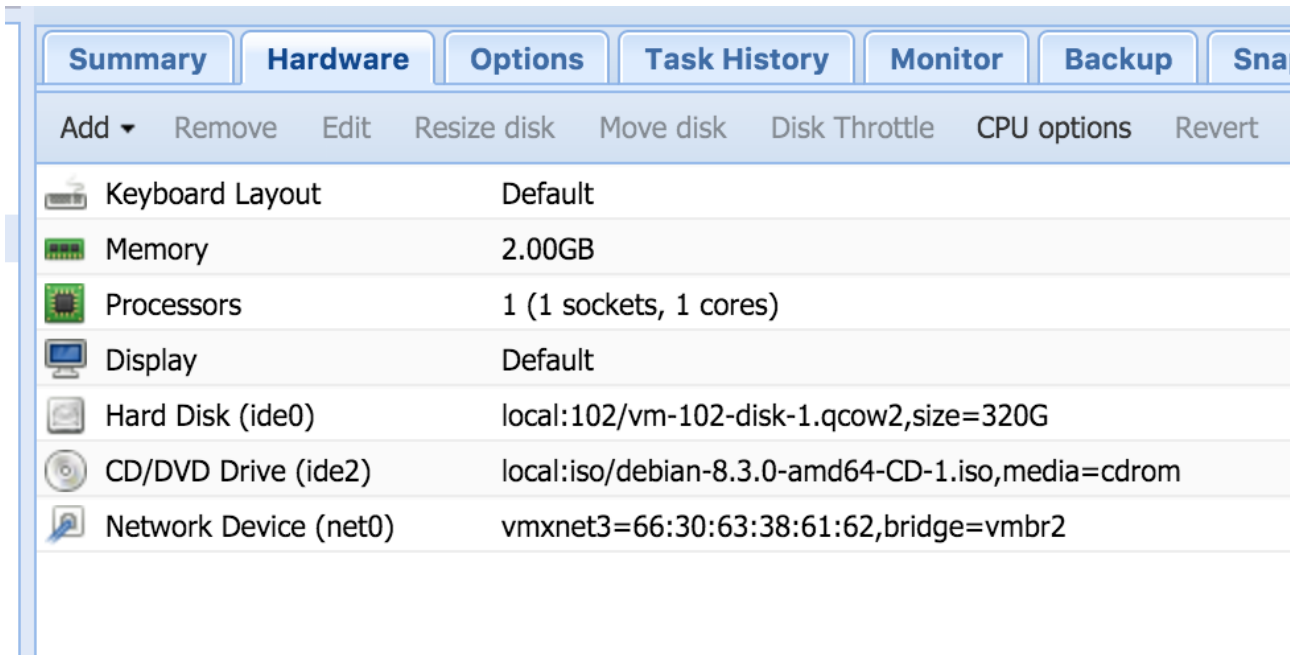
```

New VM in “Virtual LAN” Configuration

So now we almost have a full setup. Lets install this new VM in Proxmox that will sit on our new “Trust/LAN” network.

Download a Debian ISO (or whatever os you want to use) to:
`/var/lib/vz/template/iso`

Now create the VM in Proxmox, ensuring you select the vmbr2 interface.



The screenshot shows the Proxmox VM configuration interface. The 'Hardware' tab is selected. The configuration table lists various hardware components and their settings.

Component	Value
Keyboard Layout	Default
Memory	2.00GB
Processors	1 (1 sockets, 1 cores)
Display	Default
Hard Disk (ide0)	local:102/vm-102-disk-1.qcow2,size=320G
CD/DVD Drive (ide2)	local:iso/debian-8.3.0-amd64-CD-1.iso,media=cdrom
Network Device (net0)	vmxnet3=66:30:63:38:61:62,bridge=vmbr2

If you have setup DHCP then the install won't require you to enter in an IP address.

It should now be on the 10.9.2.x network. Wooo.

Juniper IPSec Home Side

Finally..... We now just need to create the site-to-site VPN on the Juniper side. This will allow us to connect directly to the new VM above.

For this we will use a route based VPN, this is because route based VPNs are cool and policy based VPNs are awful.

A route based VPN binds the VPN tunnel to a virtual interface (for example st0.0). You can then static route subnets over this new interface. It also allows for things such as BGP etc to run over it.

You just treat it like a normal network interface.

JunOS is very good at IPSec route based site-to-site VPNs, I use them a lot.

I wish some of JunOS' other features were as good as it's IPSec VPNs.

I'm just going to post the config I'm using below but the basic overview is.

- Create an unnumbered st0 interface
- Assign this interface to a new security zone
- Setup IKE & IPSec Policies
- Setup Security Policies between your trust zone and your new VPN zone
- Setup static routes
-
- Profit?

Juniper Configuration

The new ST0 interface:

```
root@jbox# edit interfaces st0

[edit interfaces st0]
root@jbox# show
unit 5 {
    description "VYOS VPN";
    family inet;
}
```

The new Security Zone with the VPN interface added to it.

```
root@jbox# edit security zones security-zone vpn

[edit security zones security-zone vpn]
root@jbox# show
interfaces {
    st0.5;
}
```

This is the IKE configuration, notice “general-ikeid” in the gateway section, this allows the VPN to work even though the VYOS box will be sending the local WAN IP instead of the public IP address.

```
[edit security ike]
root@jbox# show
proposal pre-g2-aes128-sha {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 28800;
}

policy VYOS {
    mode main;
    proposals pre-g2-aes128-sha;
    pre-shared-key ascii-text "xxxxxx"; ## SECRET-DATA
}

gateway VYOS {
    ike-policy VYOS;
    address dedicated_server_public_ip_address_here;
    external-interface pp0.0;
    general-ikeid;
}
```

This is the IPsec configuration

```
root@jbox# show
proposal g2-esp-aes128-sha {
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 3600;
}

policy VYOS {
    perfect-forward-secrecy {
        keys group2;
    }
    proposals g2-esp-aes128-sha;
}

vpn VYOS {
    bind-interface st0.5;
    ike {
        gateway VYOS;
        proxy-identity {
            local 10.0.0.0/22;
            remote 10.9.0.0/16;
        }
        ipsec-policy VYOS;
    }
    establish-tunnels immediately;
}
```

This is the security policy allowing you to connect to the external subnet.

```
[edit security policies from-zone trust to-zone vpn]
root@jbox# show
policy trust-to-vpn {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

And finally the routing options. The pp0.0 route is my internet connection.

```
[edit routing-options]
root@jbox# show
static {
    route 0.0.0.0/0 next-hop pp0.0;
    route 10.9.0.0/16 next-hop st0.5;
}
```


To Do (or not)

- Cacti Monitoring on VYOS (already setup just not documented)
- Something productive with one of the VMs (offsite backup?)
- Attempt not to run Seti on the server (probably a bad idea for everyone involved)
- Don't get DDOS?
- Improve security policies on VYOS
- Tweak Proxmox interfaces file to be better

Conclusions

- Debian is very cool
- Proxmox is cool
- VYOS is mostly cool
- JunOS is awesome and awful at the same time, so cool.
- Cheap servers are fun, but often very hot.
- I don't really like iptables, so there.
- ??