QubesOS articles

Homeserver

Author:
Neowutran

Contents

1. GOAL

THIS IS A WIP, AND I NEED HELP AND SUGGESTION TO IMPROVE IT.

- Setup a homeserver
- Use QubesOS to make the homeserver resonably secure

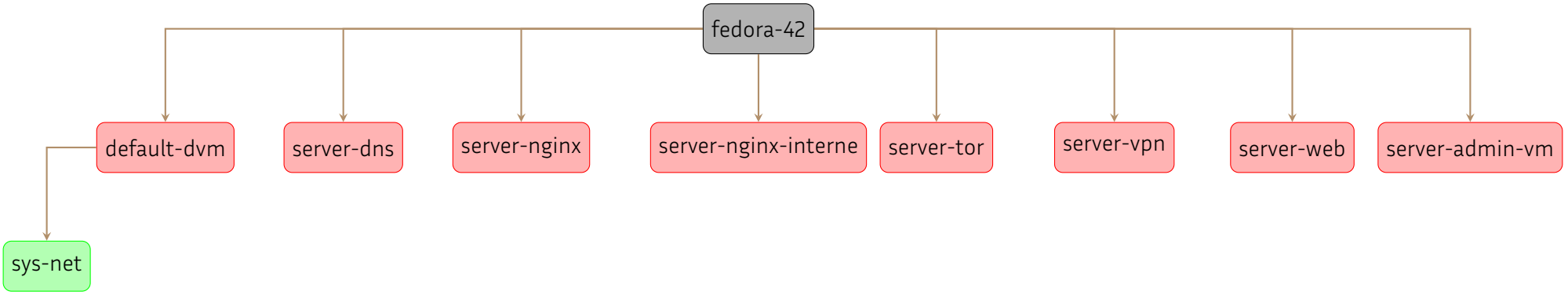
2. SERVICES

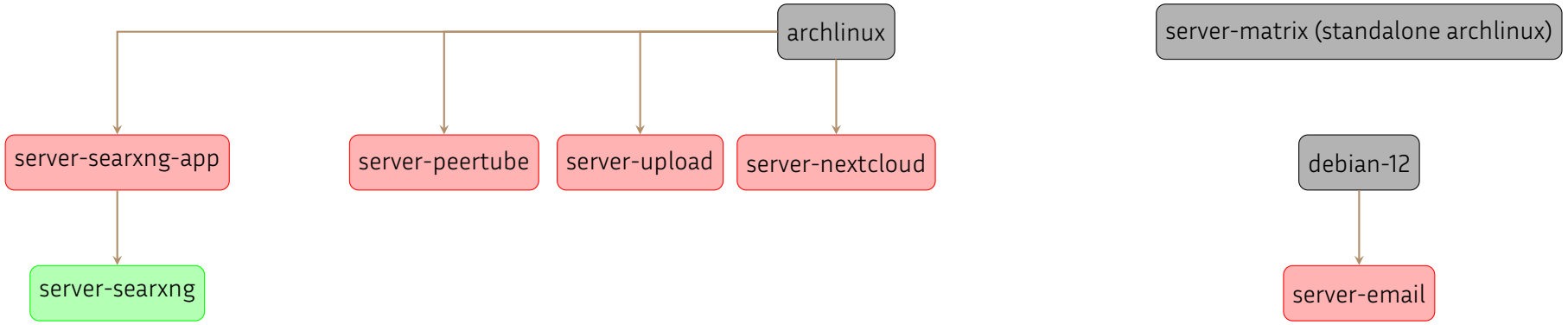
I want to host the following services:

- Email server
- Autoritative and recursive DNS server + DoH server
- Web server
- Onion service
- Searxng server
- Peertube server
- Public Nextcloud server
- VPN server
- Private Nextcloud server (accessible only by people connected to the VPN)
- Admin VM to remotely administer others server VM

3. TEMPLATES ARCHITECTURE

- Black: Template and Standalone
- Red: Application
- Green: Disposable


fedora-42
default-dvm
server-dns
server-nginx
server-nginx-interne
server-tor
server-vpn
server-web
server-admin-vm
sys-net


archlinux
server-searxng-app
server-peertube
server-upload
server-nextcloud
server-searxng
server-matrix (standalone archlinux)
debian-12
server-email

- I prefer to use fedora by default (less "bleeding edge")
- Peertube, Nextcloud and Searxng packages exist for archlinux but not for fedora/debian, so using archlinux for them.
- Using a standalone archlinux for server-matrix because I install software/packages outside official repos, and don't want to taint my archlinux template
- Server-searxng and sys-net can be disposable VM because they don't have to store and modify data files.
- Using a debian-12 template for email server because some packages I wanted are available in the official repo of debian but not fedora.

4. CHOICES I MADE

4.1 Where to put firewall rules

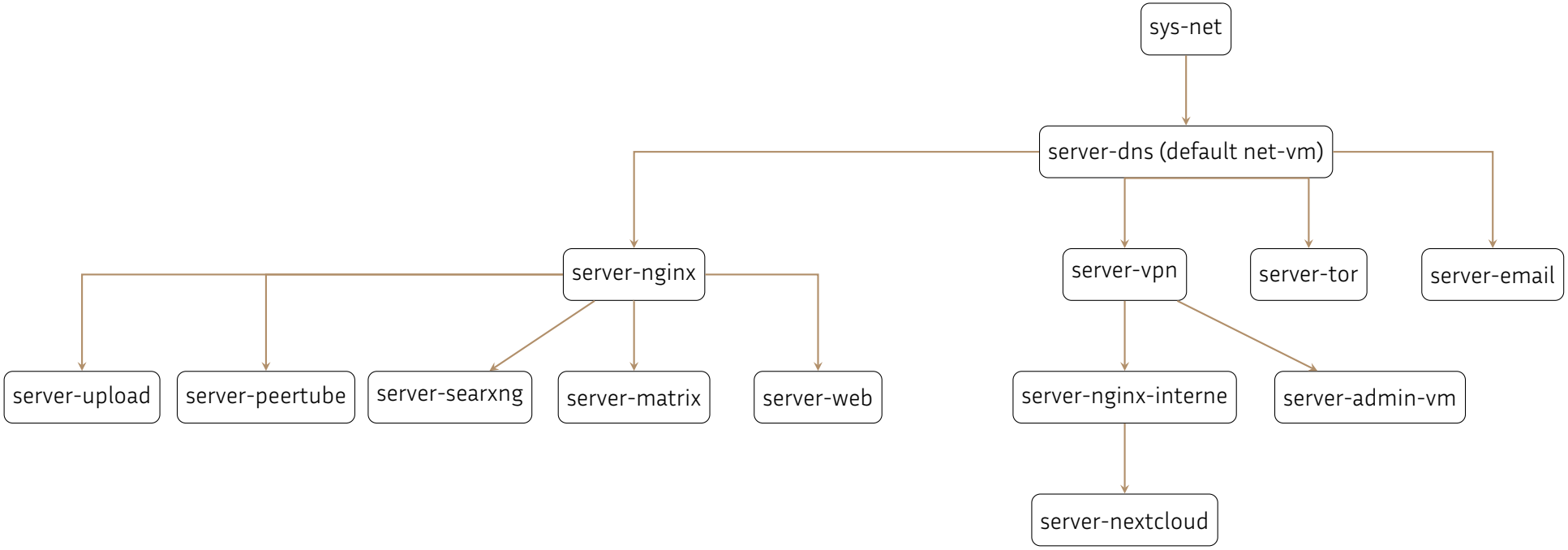
QubesOS documentation state that firewall rules should be inside /rw/config/qubes-firewall-user-script in some cases (QubesOS firewall documentation)

I make the choice to not respect that, afaik no one use this file, not even QubesOS default configuration. And this file is never actually used (or if it is used, it require more than just "For service qubes supplying networking")

4.2 qubes.ConnectTCP

I choose to not use the "qubes.ConnectTCP" policy (qubes.ConnectTCP). For my non trivial case, it feel like a better idea to use network routing instead of qrexec.

5. NETWORK ARCHITECTURE



6. NETWORK RESTRICTIONS

6.1 server-nginx-interne

Incoming

- Port 443 and only from VPN users

Outgoing

- None (other that responding to incoming request)

```
qvm-firewall server-nginx-interne
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   drop    -     -         -        -               -          -       -
```

6.2 server-upload

Incoming

- Port 80: for server-nginx

Outgoing

- None (other that responding to incoming request)

```
qvm-firewall server-upload
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   drop    -     -         -        -               -          -       -
```

6.3 server-peertube

Incoming

- Port 80: for server-nginx
- Port 9000: for server-nginx

Outgoing

- Any (for downloading video from peertube interface)

```
qvm-firewall server-peertube
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.4 server-searxng

Incoming

- Port 80: for server-nginx

Outgoing

- Any (for actually doing search queries)

```
qvm-firewall server-searxng
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.5 server-matrix

Incoming

- Port 8008: for server-nginx

Outgoing

- Any

```
qvm-firewall server-matrix
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.6 server-nginx

Incoming

- Port 443: for anyone
- Port 80: for anyone
- Port 8448: for anyone

Outgoing

- Any (no reason except some VM connected to this network VM need to access internet unrestricted.)

```
qvm-firewall server-nginx
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.7 server-vpn

Incoming

- Port 443 (udp): for anyone

Outgoing

- Any (no reason except VPN client want to access internet.)

```
qvm-firewall server-vpn
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.8 server-web

Incoming

- Port 80: for server-nginx

Outgoing

- None (other that responding to incoming request)

```
qvm-firewall server-web
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   drop    -     -         -        -               -          -       -
```

6.9 server-nextcloud

Incoming

- Port 80: for server-nginx-interne

Outgoing

- None (other that responding to incoming request)

```
qvm-firewall server-nextcloud
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   drop    -     -         -        -               -          -       -
```

6.10 server-admin-vm

Incoming

- Port 22: for server-vpn (ssh)

Outgoing

- None (other that responding to incoming request)

```
qvm-firewall server-admin-vm
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   drop    -     -         -        -               -          -       -
```

6.11 server-tor

Incoming

- None

Outgoing

- Any (outgoing traffic will be used to smuggle incoming traffic by tor)

```
qvm-firewall server-tor
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

6.12 server-email

Incoming

- SMTP port 25: for anyone (server to server communication)
- POPS port 995 : only for internal user (VPN and others local qubes)
- SMTPS port 587: only for internal user (VPN and others local qubes)

Outgoing

- HTTPS port 443: Any target (used for DANE and certbot)
- SMTP port 25: Any target

```
qvm-firewall server-email
```

```
NO  ACTION  HOST                          PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE
↪  EXPIRE  COMMENT
0   accept  -                             tcp       25       -               -          -
↪  -
1   accept  acme-v02.api.letsencrypt.org  tcp       443      -               -          -
↪  -
2   accept  -                             tcp       443      -               -          -
↪  -
3   accept  -                             -         -        dns             -          -
↪  -
4   accept  -                             icmp      -        -               -          -
↪  -
5   drop    -                             -         -        -               -          -
↪  -
```

6.13 server-dns

Incoming

- DNS port 53: for anyone
- DoH port 3053 : for server-nginx

Outgoing

- Any

```
qvm-firewall server-dns
```

```
NO  ACTION  HOST  PROTOCOL  PORT(S)  SPECIAL TARGET  ICMP TYPE  EXPIRE  COMMENT
0   accept  -     -         -        -               -          -       -
```

7. QUBES POLICY RESTRICTIONS

```
qubes.PdfConvert        *           @tag:server          @dispvm     deny
qubes.GetImageRGBA      *           @tag:server          @dispvm     deny
qubes.OpenInVM          *           @tag:server          @dispvm     deny
qubes.OpenURL           *           @tag:server          @dispvm     deny
qubes.StartApp          *           @tag:server          @dispvm     deny
qubes.SyncAppMenus      *           @tag:server          dom0        deny
qubes.UpdatesProxy      *           @tag:serve           @default    deny
qubes.VMShell           *           @tag:server          @dispvm     deny
qubes.VMExec            *           @tag:server          @dispvm     deny
qubes.VMExecGUI         *           @tag:server          @dispvm     deny
qvc.ScreenShare         *           @tag:server          @dispvm     deny


qubes.GetImageRGBA      *           @tag:server          @anyvm      deny
qubes.OpenInVM          *           @tag:server          @anyvm      deny
qubes.OpenURL           *           @tag:server          @anyvm      deny
qubes.StartApp          *           @tag:server          @anyvm      deny

qvc.ScreenShare         *           @tag:server          @adminvm    deny
qvc.ScreenShare         *           @tag:server          @anyvm      deny
qvc.Webcam              *           @tag:server          @adminvm    deny
qvc.Webcam              *           @tag:server          @default    deny
qvc.Webcam              *           @tag:server          sys-usb     deny

admin.vm.property.Get  +ip          @tag:server     @tag:server  allow   target=dom0

admin.vm.CurrentState *   @tag:server     @tag:server             allow   target=dom0
admin.vm.List         *   @tag:server     @tag:server             allow   target=dom0
admin.vm.CurrentState *   @tag:server     @adminvm                allow   target=dom0
admin.vm.List         *   @tag:server     @adminvm                allow   target=dom0


qubes.VMShell           *           server-admin-vm          @tag:server     allow
qubes.VMRootShell             *           server-admin-vm          @tag:server     allow
qubes.VMExec            *           server-admin-vm          @tag:server     allow
qubes.VMExecGUI         *           server-admin-vm          @tag:server     allow
qubes.Filecopy          *           server-admin-vm          @tag:server     allow
```

```
qvm-tags server-email add server
# ...
```

8. SCRIPTS

8.1 Dom0

Gist qvm-portfwd-nftables

```
#!/bin/bash

# Neowutran <github@neowutran.ovh>
# Adapted previous work to support QubesOS v4.2

# Adapted from previous work:
# - https://gist.github.com/fepitre/941d7161ae1150d90e15f778027e3248
# - https://gist.github.com/daktak/f887352d564b54f9e529404cc0eb60d5
# - https://gist.github.com/jpouellet/48cd0eb8589a5b9bf0c53a28fc530369
# - https://gist.github.com/Joeviocoe/6c4dc0c283f6d6c5b1a3f5af8793292b

[ "$DEBUG" = 1 ] && set -x

ip() {
        qvm-prefs -g -- "$1" ip
}

netvm() {
        qvm-prefs -g -- "$1" netvm
}

forward() {
        local action="$1"
        local from_qube="$2"
        local to_qube="$3"
        local port="$4"
        local proto="$5"
        local persistent="$6"

        local iface
        local from_ip
        local to_ip
        local nft_cmd
        local nft_handle

        # TODO: Handle multiple interfaces in sys-net. It currently catches only the first physical interface which is UP.
        iface=$(qvm-run -p -u root "$from_qube" "ip link | grep -E '^[0-9]' | grep -E 'state UP'  | cut -d ':' -f 2 | cut
        ↪  -d ' ' -f 2 | grep -vE '^(vif|lo)' | head -1")
        #from_ip=$(qvm-run -p -u root "$from_qube" "hostname -I | cut -d ' ' -f 1")
        from_ip=$(qvm-run -p -u root "$from_qube" "ip addr show dev $iface | grep -Eo 'inet [0-9]+(\.[0-9]+){3}' | cut -d '
        ↪  ' -f 2 | head -1")
        to_ip=$(ip "$to_qube")

        if [ "$from_ip" = "None" ]; then
                local from_ip=
        fi

        if [[ "$action" = "clear" ]]; then
                echo "$from_qube: Clearing Port Forwarding from $from_qube nft" >&2
                qvm-run -p -u root "$from_qube" 'data=$(nft list ruleset | grep -v '"'PortFwd $from_qube'"'); nft flush
                ↪  ruleset; echo "$data" | nft -f -'

                nft_cmd="nft list table ip qubes-firewall -a | tr -d '\"' | grep 'iifname $iface accept # handle' | awk
                ↪  '{print \$NF}'"
                nft_handle=$(qvm-run -p -u root "$from_qube" "$nft_cmd")

                if [[ $nft_handle =~ ^[0-9]+$ ]]; then
                        qvm-run -p -u root "$from_qube" "nft delete rule ip qubes-firewall forward handle $nft_handle"
                fi
                qvm-run -p -u root "$from_qube" "sed -i '/PortFwd $from_qube $to_qube:$proto$port/d' /rw/config/rc.local"
                qvm-run -p -u root "$from_qube" "sed -i '/PortFwd $from_qube $to_qube:$proto$port/d' /rw/config/rc.local"
                if ! qvm-run -p -u root "$from_qube" "grep -q 'PortFwd' /rw/config/rc.local"; then
                        qvm-run -p -u root "$from_qube" "sed -i '/nft add rule ip qubes-firewall forward meta iifname
                        ↪  $iface accept/d' /rw/config/rc.local"
                fi
        else
                echo "$from_qube: Forwarding on $iface port $port to $to_qube ($from_ip -> $to_ip)" >&2

                forward_rule1="nft -- create chain ip qubes-firewall custom-prerouting { type nat hook prerouting priority
                ↪  dstnat -1 \; }"

                forward_rule2="nft add rule ip qubes-firewall custom-prerouting iifname $iface ip daddr $from_ip $proto
                ↪  dport $port counter dnat to $to_ip "
                forward_rule2="$forward_rule2 comment \\\"PortFwd $from_qube $to_qube:$proto$port \\\" "

                forward_rule3="nft add rule ip qubes custom-forward iifname $iface ip daddr $to_ip $proto dport $port ct
                ↪  state new counter accept "
                forward_rule3="$forward_rule3 comment \\\"PortFwd $from_qube $to_qube:$proto$port\\\" "

                qvm-run -p -u root "$from_qube" 'data=$(nft list ruleset | grep -v '"'PortFwd $from_qube
                ↪  $to_qube:$proto$port'"'); nft flush ruleset; echo "$data" | nft -f -'

                qvm-run -p -u root "$from_qube" "$forward_rule1"
                qvm-run -p -u root "$from_qube" "$forward_rule2"
                qvm-run -p -u root "$from_qube" "$forward_rule3"
                if [ "$persistent" = 1 ]; then
                        qvm-run -p -u root "$from_qube" "echo $forward_rule1 >> /rw/config/rc.local"
                        qvm-run -p -u root "$from_qube" "echo $forward_rule2 >> /rw/config/rc.local"
                        qvm-run -p -u root "$from_qube" "echo $forward_rule3 >> /rw/config/rc.local"
                        if ! qvm-run -p -u root "$from_qube" "grep -q 'nft add rule ip qubes-firewall forward meta iifname
                        ↪  $iface accept' /rw/config/rc.local"; then
                                # Ensure rc.local is executable
                                qvm-run -p -u root "$from_qube" "chmod +x /rw/config/rc.local"
                        fi
                fi
        fi
}
```

```
input() {
        local action="$1"
        local qube="$2"
        local port="$3"
        local proto="$4"
        local persistent="$5"

        if [[ "$action" = "clear" ]]; then
                echo "$qube: Clearing Port Forwarding from $qube nft" >&2
                qvm-run -p -u root "$qube" 'data=$(nft list ruleset | grep -v '"'PortFwd $qube'"'); nft flush ruleset; echo
                ↪  "$data" | nft -f -'
                qvm-run -p -u root "$qube" "sed -i '/PortFwd $qube:$proto$port/d' /rw/config/rc.local"
        else
                echo "$qube: Allowing input to port $port" >&2
                qvm-run -p -u root "$qube" 'data=$(nft list ruleset | grep -v '"'PortFwd $qube:$proto$port'"'); nft flush
                ↪  ruleset; echo "$data" | nft -f -'


                input_rule="nft add rule ip qubes custom-input $proto dport $port ct state new counter accept comment
                ↪  \\\"PortFwd $qube:$proto$port\\\" "
                qvm-run -p -u root "$qube" "$input_rule"
                if [ "$persistent" = 1 ]; then
                        qvm-run -p -u root "$qube" "echo $input_rule >> /rw/config/rc.local"
                        # Ensure rc.local is executable
                        qvm-run -p -u root "$qube" "chmod +x /rw/config/rc.local"
                fi
        fi
}

recurse_netvms() {
        local action="$1"
        local this_qube="$2"
        local port="$3"
        local proto="$4"
        local persistent="$5"

        local outer_dom

        outer_dom=$(netvm "$this_qube")
        if [[ -n "$outer_dom" && "$outer_dom" != "None" ]]; then
                forward "$action" "$outer_dom" "$this_qube" "$port" "$proto" "$persistent"
                recurse_netvms "$action" "$outer_dom" "$port" "$proto" "$persistent"
        fi
}

usage() {
        echo "Usage: ${0##*/} --action ACTION --qube QUBE --port PORT --proto PROTO --persistent" >&2
        echo "" >&2
        echo "Exemple: " >&2
        echo " -> ${0##*/} --action create --qube work --port 22" >&2
        echo " -> ${0##*/} --action create --qube work --port 444 --proto udp --persistent" >&2
        echo " -> ${0##*/} --action clear --qube work --port 22" >&2
        echo " -> ${0##*/} --action clear --qube work --port 444 --proto udp" >&2
        echo "" >&2
        echo "Default value for PROTO is 'tcp' and nft are not persistent"
        exit 1
}

if ! OPTS=$(getopt -o a:q:p:n:s --long action:,qube:,port:,proto:,persistent -n "$0" -- "$@"); then
        echo "An error occurred while parsing options." >&2
        exit 1
fi

eval set -- "$OPTS"

while [[ $# -gt 0 ]]; do
        case "$1" in
                -a | --action ) ACTION="$2"; shift ;;
                -q | --qube ) QUBE="$2"; shift ;;
                -p | --port ) PORT="$2"; shift ;;
                -n | --proto ) PROTO="$2"; shift ;;
                -s | --persistent ) PERSISTENT=1; shift ;;
        esac
        shift
done

if [ -z "$PROTO" ]; then
        PROTO="tcp"
fi

if { [ "$ACTION" != "create" ] || [ "$ACTION" == "clear" ]; } && { [ -z "$QUBE" ] || [ -z "$PORT" ]; }; then
        usage
fi

if ! qvm-check "$QUBE" > /dev/null 2>&1; then
        echo "Qube '$QUBE' not found." >&2
        exit 1
fi

input "$ACTION" "$QUBE" "$PORT" "$PROTO" "$PERSISTENT"
recurse_netvms "$ACTION" "$QUBE" "$PORT" "$PROTO" "$PERSISTENT"
```

monthly.sh

```
#!/bin/bash

qvm-run -u root --pass-io server-nginx "cd /home/user/; echo '2'
↪  | ./certs.sh"
qvm-run -u root --pass-io server-email "cd /home/user/; echo '2'
↪  | ./certs.sh"
qvm-run -u root --pass-io wildcard_certificate 'echo "2" |
↪  certbot certonly --manual -d neowutran.ovh -d
↪  "*.neowutran.ovh" --preferred-challenges dns-01 --server
↪  https://acme-v02.api.letsencrypt.org/directory'
certs=$(qvm-run -u root --pass-io wildcard_certificate 'tar
↪  -czvO /etc/letsencrypt/archive/ /etc/letsencrypt/live/ |
↪  base64 -w 0')
qvm-shutdown wildcard_certificate
qvm-run --pass-io server-nginx-interne "echo -n '$certs' |
↪  base64 -d > /root/certs.tar.gz"
qvm-run -u root --pass-io server-nginx-interne "cd /;tar -xzvf
↪  /root/certs.tar.gz; rm /root/certs.tar.gz"
qvm-run -u root --pass-io server-nginx-interne "systemctl
↪  restart nginx"
qvm-run -u root --pass-io server-nginx "systemctl restart nginx"

dane1=$(qvm-run -u root --pass-io server-nginx "cd /home/user/;
↪  ./generate_dane.sh | base64 -w 0")
dane2=$(qvm-run -u root --pass-io server-email "cd /home/user/;
↪  ./generate_dane.sh | base64 -w 0")
dane3=$(qvm-run -u root --pass-io server-nginx-interne "cd
↪  /home/user/; ./generate_dane.sh | base64 -w 0")

qvm-run -u root --pass-io server-dns "cd /home/user/dnssec; cp
↪  base_neowutran.ovh neowutran.ovh.zone"
qvm-run -u root --pass-io server-dns "cd /home/user/dnssec; echo
↪  '$dane1' | base64 -d >> neowutran.ovh.zone"
qvm-run -u root --pass-io server-dns "cd /home/user/dnssec; echo
↪  '$dane2' | base64 -d >> neowutran.ovh.zone"
qvm-run -u root --pass-io server-dns "cd /home/user/dnssec; echo
↪  '$dane3' | base64 -d >> neowutran.ovh.zone"
qvm-run -u root --pass-io server-dns "cd /home/user/dnssec;
↪  ./sign.sh"

qvm-run -u root --pass-io server-email "systemctl restart
↪  postfix; systemctl restart dovecot"
```

monthly.service

```
[Unit]
Description=Run monthly.sh script

[Service]
Type=simple
ExecStart=/bin/bash /home/user/monthly.sh
User=user
Group=user
```

monthly.timer

```
[Unit]
Description=Run monthly.sh script monthly

[Timer]
OnCalendar=monthly
Persistent=true

[Install]
WantedBy=timers.target
```

servers.sh

```
#!/bin/bash

#xenpm enable-turbo-mode
#xenpm set-scaling-governor performance
/usr/bin/sleep 20

/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 80 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 443 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 8448 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 3478 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 3478 --proto udp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 5349 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 5349 --proto udp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-nginx --port 49152-65535 --proto udp

/home/user/qvm-portfwd-nftables --action create --qube
↪  server-email --port 25 --proto tcp

/home/user/qvm-portfwd-nftables --action create --qube
↪  server-vpn --port 443 --proto udp

/home/user/qvm-portfwd-nftables --action create --qube
↪  server-gl0 --port 2222 --proto tcp

/home/user/qvm-portfwd-nftables --action create --qube
↪  server-dns --port 53 --proto tcp
/home/user/qvm-portfwd-nftables --action create --qube
↪  server-dns --port 53 --proto udp


/usr/bin/qvm-start server-web
/usr/bin/qvm-start server-matrix
/usr/bin/qvm-start server-peertube
/usr/bin/qvm-start server-searxng
/usr/bin/qvm-start server-upload
/usr/bin/qvm-start server-gl0
/usr/bin/qvm-start server-element
/usr/bin/qvm-start server-nextcloud
/usr/bin/qvm-start server-tor
/usr/bin/qvm-start server-admin-vm
```

startup.service

```
[Unit]
Description=Run servers.sh script
After=network.target

[Service]
Type=simple
ExecStart=/bin/bash /home/user/servers.sh
User=user
Group=user

[Install]
WantedBy=multi-user.target
```

8.2 server-tor

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/tor/torrc' )
binds+=( '/var/lib/tor/' )
binds+=( '/etc/nginx/' )
```

/rw/config/rc.local

```
#!/bin/sh

/usr/sbin/setsebool httpd_can_network_connect 1 -P
/usr/sbin/chown -R toranon:toranon /var/lib/tor/
/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start nginx
/usr/sbin/systemctl enable tor
/usr/sbin/systemctl start tor
```

/etc/nginx/nginx.conf (partial extract)

```
server {
 listen localhost:80;
 server_name zv4geh5hu5kdoijeqyj6fxfssjcawttqpdwlzapzc5cclu2il
  ↪  m2zphid.onion;
 location / {
        # Server-nginx
  proxy_pass https://10.137.0.69:443;
  proxy_set_header   Host             neowutran.ovh;
 }
}
```

8.3 server-web

/rw/config/rc.local

```
#!/bin/sh

# allow server-nginx
server_nginx=$(/usr/sbin/python3 /usr/lib/python3.13/site-packa
 ↪  ges/qubesadmin/tools/qvm_prefs.py server-nginx ip)
/usr/sbin/nft add rule ip qubes custom-input ip saddr
 ↪  "$server_nginx" tcp dport 80 ct state new counter accept

/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start nginx
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/nginx/' )
binds+=( '/var/www/html/' )
```

8.4 server-vpn

/rw/config/rc.local

```
#!/bin/sh

/usr/sbin/sysctl -w net.ipv4.conf.all.route_localnet=1
/usr/sbin/sysctl -w net.ipv4.ip_forward=1

/usr/sbin/systemctl enable wg-quick@server
/usr/sbin/systemctl start wg-quick@server
/usr/sbin/nft add rule ip qubes custom-forward 'iif server
↪  counter accept'
echo module wireguard +p >
↪  /sys/kernel/debug/dynamic_debug/control

/usr/sbin/nft -- create chain ip qubes custom-prerouting { type
↪  nat hook prerouting priority dstnat -1 \; }

server_email=$(/usr/sbin/python3 /usr/lib/python3.13/site-packa
↪  ges/qubesadmin/tools/qvm_prefs.py server_email ip)
/usr/sbin/nft add rule ip qubes custom-prerouting ip protocol
↪  tcp ip saddr == 10.200.200.1/24 ip daddr 10.200.200.1 tcp
↪  dport 995 counter dnat to "$server_email":995
/usr/sbin/nft add rule ip qubes custom-prerouting ip protocol
↪  tcp ip saddr == 10.200.200.1/24 ip daddr 10.200.200.1 tcp
↪  dport 587 counter dnat to "$server_email":587

server_nginx_interne=$(/usr/sbin/python3 /usr/lib/python3.13/si
↪  te-packages/qubesadmin/tools/qvm_prefs.py
↪  server-nginx-interne ip)
/usr/sbin/nft add rule ip qubes custom-prerouting ip protocol
↪  tcp ip saddr == 10.200.200.1/24 ip daddr 10.200.200.1 tcp
↪  dport 443 counter dnat to "$server_nginx_interne":443

server_dns=$(/usr/sbin/python3 /usr/lib/python3.13/site-package
↪  s/qubesadmin/tools/qvm_prefs.py server-dns ip)
/usr/sbin/nft add rule ip qubes custom-prerouting ip protocol
↪  udp ip saddr == 10.200.200.1/24 ip daddr 10.200.200.1 udp
↪  dport 53 counter dnat to "$server_dns":53

server_admin_vm=$(/usr/sbin/python3 /usr/lib/python3.13/site-pa
↪  ckages/qubesadmin/tools/qvm_prefs.py server-admin-vm ip)
/usr/sbin/nft add rule ip qubes custom-prerouting ip protocol
↪  tcp ip saddr == 10.200.200.5 ip daddr 10.200.200.1 tcp
↪  dport 22 counter dnat to "$server_admin_vm":22

sys_vpn=$(/usr/sbin/python3 /usr/lib/python3.13/site-packages/q
↪  ubesadmin/tools/qvm_prefs.py sys-vpn ip)
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  "$sys_vpn" udp dport 443 ct state new counter accept
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/wireguard/' )
```

8.5 server-nginx-interne

certs.sh

```
#!/bin/bash
/usr/sbin/certbot certonly --manual -d neowutran.ovh -d
↪  '*.neowutran.ovh' --preferred-challenges dns-01 --server
↪  https://acme-v02.api.letsencrypt.org/directory
```

generate_dane.sh

```
hash1=$(/usr/sbin/openssl x509 -in
↪  /etc/letsencrypt/live/neowutran.ovh/cert.pem -noout -pubkey
↪  | /usr/sbin/openssl pkey -pubin -outform DER |
↪  /usr/sbin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f
↪  2 )
cd /root/
cat /etc/letsencrypt/live/neowutran.ovh/fullchain.pem | awk
↪  'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "bundlecert." c
↪  ".crt"}'
hash2=$(/usr/sbin/openssl x509 -in bundlecert.2.crt -noout
↪  -pubkey | /usr/sbin/openssl pkey -pubin -outform DER |
↪  /usr/sbin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f
↪  2)

for domain in nextcloud.neowutran.ovh
do
echo "_443._tcp.$domain. IN TLSA 3 1 1 $hash1"
echo "_443._tcp.$domain. IN TLSA 2 1 1 $hash2"
done
```

/rw/config/rc.local

```
/usr/sbin/setsebool httpd_can_network_connect 1 -P

# Allow VPN user
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.200.200.1/24 tcp dport 443 ct state new counter accept

/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start nginx
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/nginx/nginx.conf' )
binds+=( '/etc/letsencrypt/' )
```

8.6 server-email

certs.sh

```
#!/bin/bash
/usr/sbin/certbot certonly --manual -d email.neowutran.ovh
↪  --preferred-challenges dns-01 --server
↪  https://acme-v02.api.letsencrypt.org/directory
```

generate_dane.sh

```
#!/bin/bash
hash1=$(/usr/bin/openssl x509 -in
↪  /etc/letsencrypt/live/email.neowutran.ovh/cert.pem -noout
↪  -pubkey | /usr/bin/openssl pkey -pubin -outform DER |
↪  /usr/bin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f 2
↪  )
echo "_25._tcp.email.neowutran.ovh. IN TLSA 3 1 1 $hash1"

cd /root/
cat /etc/letsencrypt/live/email.neowutran.ovh/fullchain.pem |
↪  awk 'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "bundlecert."
↪  c ".crt"}'
hash2=$(/usr/bin/openssl x509 -in bundlecert.2.crt -noout
↪  -pubkey | /usr/bin/openssl pkey -pubin -outform DER |
↪  /usr/bin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f 2)
echo "_25._tcp.email.neowutran.ovh. IN TLSA 2 1 1 $hash2"
```

/rw/config/rc.local

```
/usr/sbin/systemctl start opendkim
/usr/sbin/systemctl start opendmarc
/usr/sbin/systemctl start postfix
/usr/sbin/systemctl start dovecot

server_vpn=$(/usr/bin/python3 /usr/lib/python3/dist-packages/qu
↪  besadmin/tools/qvm_prefs.py server-vpn ip)
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  "$server_vpn" tcp dport 995 ct state new counter accept
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  "$server_vpn" tcp dport 587 ct state new counter accept
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/dovecot/' )
binds+=( '/etc/letsencrypt/' )
binds+=( '/etc/opendkim.conf' )
binds+=( '/etc/dkimkeys/' )
binds+=( '/etc/opendmarc.conf' )
binds+=( '/etc/postfix/' )
binds+=( '/etc/postfix-policyd-spf-python/' )
```

8.7 server-searxng

/rw/config/rc.local

```
#!/bin/sh

/usr/sbin/nft add rule ip  qubes custom-input ip saddr
↪  10.137.0.69 tcp dport 80 ct state new counter accept
/usr/sbin/systemctl enable redis
/usr/sbin/systemctl start redis
/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start nginx
/usr/sbin/mkdir /etc/searxng/
/usr/sbin/cp /home/user/settings.yml /etc/searxng/
/usr/sbin/cp /home/user/limiter.toml /etc/searxng/
/usr/sbin/cp /home/user/rewrite-hosts.yml /etc/searxng/

cd /home/user/searxng
/usr/bin/git stash
/usr/bin/git pull
/usr/sbin/sudo -H ./utils/searxng.sh install user
echo -e "\n\n\n\n\n" | /usr/sbin/sudo -H ./utils/searxng.sh
↪  instance update
echo -e "\n\n\n\n\n\n" | /usr/sbin/sudo -H ./utils/searxng.sh
↪  install uwsgi pyenv

/usr/sbin/systemctl enable uwsgi@searxng
/usr/sbin/systemctl start uwsgi@searxng
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/usr/local/searxng' )
binds+=( '/etc/nginx/nginx.conf' )
```

8.8 server-peertube

/rw/config/rc.local

```
#!/bin/sh

/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.137.0.69 tcp dport 80 ct state new counter accept
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.137.0.69 tcp dport 9000 ct state new counter accept
/usr/sbin/useradd -m -d /var/www/peertube -s /bin/bash -p
↪  peertube peertube
/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start  nginx
/usr/sbin/systemctl start  postgresql
/usr/sbin/systemctl enable postgresql
/usr/sbin/systemctl enable redis
/usr/sbin/systemctl start redis
/usr/sbin/cp /home/user/peertube.service /etc/systemd/system/
/usr/sbin/cp /home/user/30-peertube-tcp.conf /etc/sysctl.d/
/usr/sbin/sysctl -p /etc/sysctl.d/30-peertube-tcp.conf
/usr/sbin/systemctl daemon-reload
/usr/sbin/systemctl enable peertube
/usr/sbin/systemctl start peertube
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/nginx/nginx.conf' )
binds+=( '/var/lib/postgres/' )
binds+=( '/var/www/' )
```

8.9 server-matrix

/rw/config/rc.local

```
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.137.0.69 tcp dport 8008 ct state new counter accept
```

8.10 server-nextcloud

/rw/config/rc.local

```
# Allow server-nginx-interne
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.137.0.118 tcp dport 80 ct state new counter accept

export NEXTCLOUD_PHP_CONFIG=/etc/webapps/nextcloud/php.ini
/usr/bin/mkdir -p /var/log/php-legacy-fpm/access
/usr/bin/mkdir -p /var/log/php-fpm/access
/usr/bin/systemctl enable php-legacy-fpm
/usr/bin/systemctl start php-legacy-fpm
/usr/bin/systemctl enable php-fpm
/usr/bin/systemctl start php-fpm
/usr/bin/systemctl enable mariadb
/usr/bin/systemctl start mariadb
/usr/bin/systemctl enable nginx
/usr/bin/systemctl start nginx
/usr/bin/systemctl enable redis
/usr/bin/systemctl start redis
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/webapps/nextcloud/' )
binds+=( '/var/lib/nextcloud/' )
binds+=( '/etc/nginx/nginx.conf' )
binds+=( '/etc/my.cnf.d/server.cnf' )
binds+=( '/var/lib/mysql/' )
binds+=( '/etc/php-legacy/php-fpm.d/' )
binds+=( '/etc/php/php-fpm.d/' )
binds+=( '/usr/lib/systemd/system/php-fpm.service' )
```

8.11 server-upload

/rw/config/rc.local

```
# Allow server-nginx
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.137.0.69 tcp dport 80 ct state new counter accept

export NEXTCLOUD_PHP_CONFIG=/etc/webapps/nextcloud/php.ini
/usr/bin/mkdir -p /var/log/php-legacy-fpm/access
/usr/bin/mkdir -p /var/log/php-fpm/access
/usr/bin/systemctl enable php-legacy-fpm
/usr/bin/systemctl start php-legacy-fpm
/usr/bin/systemctl enable php-fpm
/usr/bin/systemctl start php-fpm
/usr/bin/systemctl enable mariadb
/usr/bin/systemctl start mariadb
/usr/bin/systemctl enable nginx
/usr/bin/systemctl start nginx
/usr/bin/systemctl enable redis
/usr/bin/systemctl start redis
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/webapps/nextcloud/' )
binds+=( '/var/lib/nextcloud/' )
binds+=( '/etc/nginx/nginx.conf' )
binds+=( '/etc/my.cnf.d/server.cnf' )
binds+=( '/var/lib/mysql/' )
binds+=( '/etc/php-legacy/php-fpm.d/' )
binds+=( '/etc/php/php-fpm.d/' )
binds+=( '/usr/lib/systemd/system/php-fpm.service' )
```

8.12 server-nginx

certs.sh

```
#!/bin/bash
/usr/sbin/certbot certonly --manual \
 -d neowutran.ovh \
 -d maisonhome.neowutran.ovh \
 -d matrix.neowutran.ovh \
 -d searxng.neowutran.ovh \
 -d mtube.neowutran.ovh \
 -d dns.neowutran.ovh \
 -d web.neowutran.ovh \
 -d element.neowutran.ovh \
 -d mta-sts.neowutran.ovh \
 -d www.neowutran.ovh \
 -d upload.neowutran.ovh \
 --preferred-challenges dns-01 --server
  ↪  https://acme-v02.api.letsencrypt.org/directory
```

generate_dane.sh

```
hash1=$(/usr/sbin/openssl x509 -in
 ↪  /etc/letsencrypt/live/neowutran.ovh/cert.pem -noout -pubkey
 ↪  | /usr/sbin/openssl pkey -pubin -outform DER |
 ↪  /usr/sbin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f
 ↪  2 )
cd /root/
cat /etc/letsencrypt/live/neowutran.ovh/fullchain.pem | awk
 ↪  'BEGIN {c=0;} /BEGIN CERT/{c++} { print > "bundlecert." c
 ↪  ".crt"}'
hash2=$(/usr/sbin/openssl x509 -in bundlecert.2.crt -noout
 ↪  -pubkey | /usr/sbin/openssl pkey -pubin -outform DER |
 ↪  /usr/sbin/openssl sha256 | cut -d '=' -f 2 | cut -d ' ' -f
 ↪  2)

for domain in searxng.neowutran.ovh mta-sts.neowutran.ovh
 ↪  neowutran.ovh dns.neowutran.ovh XXXX.neowutran.ovh
do
echo "_443._tcp.$domain. IN TLSA 3 1 1 $hash1"
echo "_443._tcp.$domain. IN TLSA 2 1 1 $hash2"
done
```

/rw/config/rc.local

```
# expose port for tor onion service VM
server_tor=$(/usr/sbin/python3 /usr/lib/python3.13/site-package
↪  s/qubesadmin/tools/qvm_prefs.py server-tor ip)
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  "server_tor" tcp dport 443 ct state new counter accept

/usr/sbin/semanage port -a -t http_port_t -p tcp 8448
/usr/sbin/setsebool httpd_can_network_connect 1 -P
/usr/sbin/cp /home/user/30-peertube-tcp.conf /etc/sysctl.d/
/usr/sbin/sysctl -p /etc/sysctl.d/30-peertube-tcp.conf
/usr/sbin/systemctl enable nginx
/usr/sbin/systemctl start nginx
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/nginx/nginx.conf' )
binds+=( '/etc/letsencrypt/' )
```

8.13 server-dns

sign.sh

```
#!/bin/bash
/usr/sbin/dnssec-signzone -A -N INCREMENT -o neowutran.ovh -t
↪  neowutran.ovh.zone
/usr/sbin/cp neowutran.ovh.zone* /etc/unbound/
/usr/sbin/systemctl restart unbound
```

/rw/config/rc.local

```
/usr/sbin/sysctl -w net.ipv4.conf.all.route_localnet=1

/usr/sbin/nft add rule ip qubes custom-input iifname "vif*" ip daddr 127.0.0.1 udp dport 53
↪  counter accept
/usr/sbin/nft add rule ip qubes custom-input iifname "vif*" ip daddr 127.0.0.1 tcp dport 53
↪  counter accept

# redirect dns-requests to localhost
/usr/sbin/nft flush chain ip qubes dnat-dns
/usr/sbin/nft add rule ip qubes dnat-dns ip daddr 10.139.1.1 udp dport 53 counter dnat to
↪  127.0.0.1
/usr/sbin/nft add rule ip qubes dnat-dns ip daddr 10.139.1.1 tcp dport 53 counter dnat to
↪  127.0.0.1
/usr/sbin/nft add rule ip qubes dnat-dns ip daddr 10.139.1.2 udp dport 53 counter dnat to
↪  127.0.0.1
/usr/sbin/nft add rule ip qubes dnat-dns ip daddr 10.139.1.2 tcp dport 53 counter dnat to
↪  127.0.0.1

/usr/sbin/systemctl disable systemd-resolved
/usr/sbin/systemctl stop systemd-resolved
echo "nameserver 127.0.0.1" > /etc/resolv.conf

/usr/sbin/cp /home/user/dnssec/neowutran.ovh.zone /etc/unbound/
/usr/sbin/cp /home/user/dnssec/neowutran.ovh.zone.signed /etc/unbound/
/usr/sbin/systemctl enable unbound
/usr/sbin/systemctl start unbound

server_nginx=$(/usr/sbin/python3
↪  /usr/lib/python3.13/site-packages/qubesadmin/tools/qvm_prefs.py server-nginx ip)
server_tor=$(/usr/sbin/python3
↪  /usr/lib/python3.13/site-packages/qubesadmin/tools/qvm_prefs.py server-tor ip)
server_email=$(/usr/sbin/python3
↪  /usr/lib/python3.13/site-packages/qubesadmin/tools/qvm_prefs.py server-email ip)
server_vpn=$(/usr/sbin/python3
↪  /usr/lib/python3.13/site-packages/qubesadmin/tools/qvm_prefs.py server-vpn ip)

/usr/sbin/nft add rule ip qubes custom-input ip saddr "$server_nginx" tcp dport 3053 ct
↪  state new counter accept
/usr/sbin/nohup /home/user/.cargo/bin/doh-proxy -l 0.0.0.0:3053 -u 127.0.0.1:53 -j 443 -g
↪  82.65.3.49 -H dns.neowutran.ovh -p / -O -C 10000 -K &


# TOR ONION SERVICE FORWARDING
/usr/sbin/nft add rule ip qubes custom-forward ip saddr "$server_tor" ip daddr
↪  "$server_nginx" tcp dport 443 ct state new,established counter accept

# Email forwarding
/usr/sbin/nft -- create chain ip qubes custom-prerouting { type nat hook prerouting
↪  priority dstnat -1 \; }

/usr/sbin/nft add rule ip qubes custom-forward ip saddr "$server_vpn" ip daddr
↪  "$server_email" tcp dport 995 ct state new,established counter accept
/usr/sbin/nft add rule ip qubes custom-forward ip saddr "$server_vpn" ip daddr
↪  "$server_email" tcp dport 587 ct state new,established counter accept
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/unbound/unbound.conf' )
```

8.14 server-admin-vm

/rw/config/rc.local

```
# TODO: trouver la configuration SELinux qui accepte SSHD avec
↪  pam_oath
setenforce 0

# Allow VPN user
/usr/sbin/nft add rule ip qubes custom-input ip saddr
↪  10.200.200.1/24 tcp dport 22 ct state new counter accept

/usr/sbin/systemctl enable sshd
/usr/sbin/systemctl start sshd
```

/rw/config/qubes-bind-dirs.d/50-user.conf

```
binds+=( '/etc/ssh/sshd_config' )
binds+=('/etc/pam.d/sshd')
binds+=('/etc/ssh/sshd_config.d/')
```

9. INTERACTIONS

Systemd service in dom0 launching every month, see "monthly.sh" for the action it does.