ADMINISTRACIÓN DE SERVIDORES GNU/LINUX
DEPARTAMENTO UNIVERSITARIO DE INFORMÁTICA
UNIVERSIDAD NACIONAL DE CÓRDOBA

PARCIAL I I
SERVIDOR PyME

Autor: Exequiel Barrirero

APELLIDO, NOMBRE

Índice

Project Repository:

Parcial-II:
https://github.com/exequielrafaela/Vagrant_PyFab-Munin-Postfix-Squid

Complementario Parcial-II
https://github.com/exequielrafaela/proton

Configuración Vagrant:

Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# Check Vagrant dependencies
if !Vagrant.has_plugin?("vagrant-vbguest")
      system('vagrant plugin install vagrant-vbguest')

     raise("vagrant-vbguest installed. Run command \"vagrant up\" again.");
end

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.require_version '>= 1.8.0'
Vagrant.configure(2) do |config|

  # Create and configure the VM
  config.vm.define :mailproxylinux do |srv|
    srv.vm.box = "ubuntu/trusty64"
    # Assign additional private network
    srv.vm.network "private_network", ip: "172.16.0.10"
    srv.vm.hostname = "mailproxylinux"
    srv.vm.synced_folder "../fabric", "/fabric", create: true
    srv.vm.synced_folder "../conf", "/conf", create: true
    srv.vm.synced_folder "../data", "/data", create: true
    # Configure CPU & RAM per settings
    srv.vm.provider "virtualbox" do |vb|
      vb.memory = 1024
      vb.cpus = 2
    end
    # Provisioner BashShell
    srv.vm.provision "shell", path: "./script.sh"
  end
end
```

Network Config output

```
==> mailproxylinux: ======================================================================
==> mailproxylinux: ## NETWORK CONFIGURATION #
==> mailproxylinux: ======================================================================
==> mailproxylinux: [localhost] sudo: ip addr show
==> mailproxylinux: 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default
==> mailproxylinux:     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
==> mailproxylinux:     inet 127.0.0.1/8 scope host lo
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 ::1/128 scope host
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux: 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
==> mailproxylinux:     link/ether 08:00:27:48:af:80 brd ff:ff:ff:ff:ff:ff
```

```
==> mailproxylinux:     inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 fe80::a00:27ff:fe48:af80/64 scope link
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux: 3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
==> mailproxylinux:     link/ether 08:00:27:e8:98:a3 brd ff:ff:ff:ff:ff:ff
==> mailproxylinux:     inet 172.16.0.10/24 brd 172.16.0.255 scope global eth1
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 fe80::a00:27ff:fee8:98a3/64 scope link
==> mailproxylinux:        valid_lft forever preferred_lft forever
```

Configuración servicio APACHE2

7.1.- Configurar apache para que solo funcione en HTTPS puerto 443.

Texto

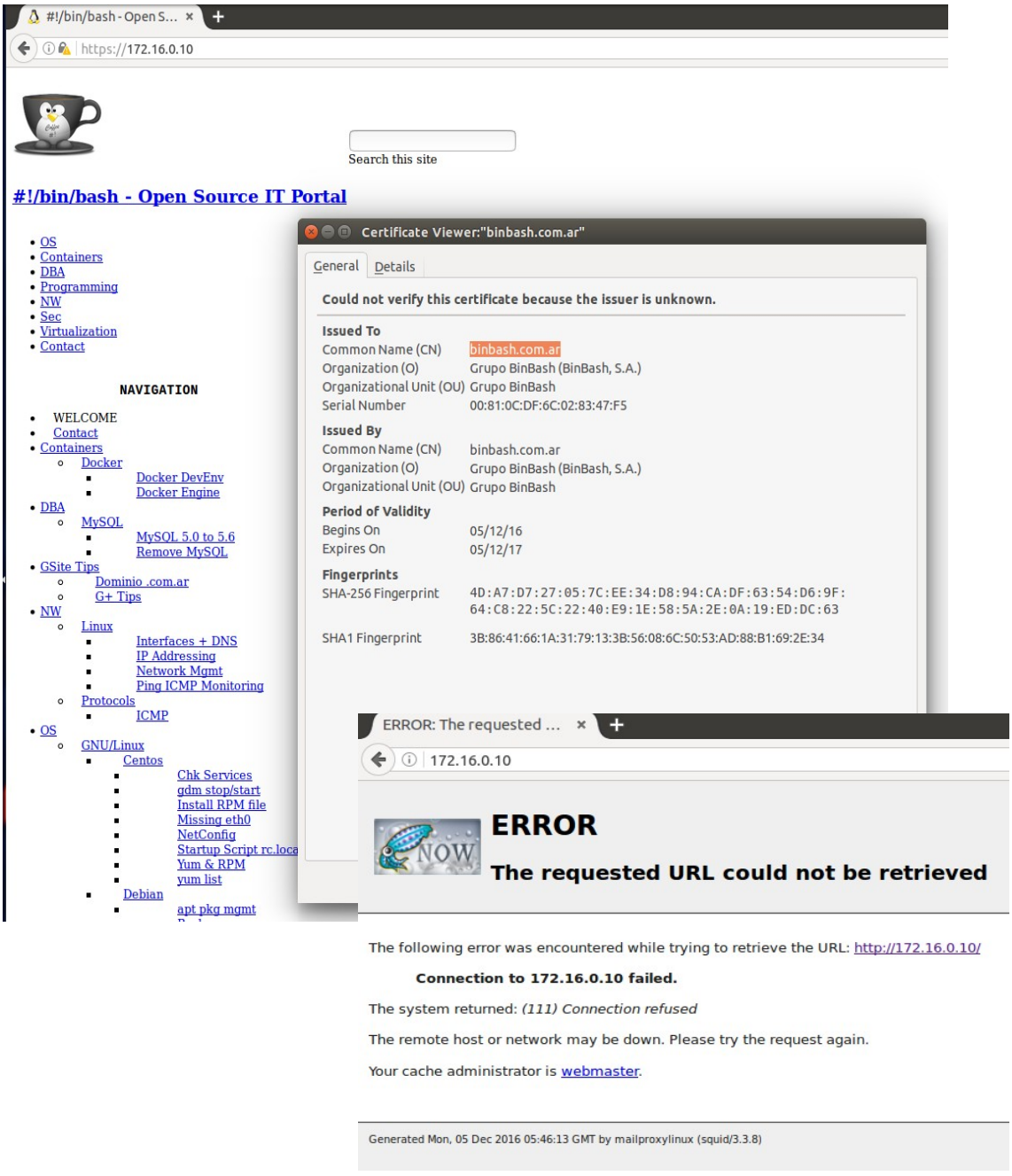
Instalación y Configuración

```
def install_apache24_ubuntu_14():
    """
Apache2 HTTP Server installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('############################', 'blue')
        print colored('#### APACHE2 WEB_SERV ####', 'blue')
        print colored('############################', 'blue')
        sudo('apt-get install -y apache2')
        sudo('sh /conf/apache2/gen-cer.sh binbash.com.ar')
        sudo('cp /conf/apache2/ports.conf /etc/apache2/ports.conf')
        sudo('cp /conf/apache2/binbash.com.ar.conf /etc/apache2/sites-
available/binbash.com.ar.conf')
        sudo('mkdir -p /var/www/binbash.com.ar/public_html')
        sudo('mkdir -p /var/www/binbash.com.ar/logs')
        sudo('wget -P /var/www/binbash.com.ar'
            ' --recursive'
            ' --no-clobber'
            ' --page-requisites'
            ' --html-extension'
            ' --convert-links'
            ' --restrict-file-names=windows'
            ' --domains website.org'
            ' --no-parent'
            ' http://www.binbash.com.ar')
        sudo('cp /var/www/binbash.com.ar/www.binbash.com.ar/index.html
/var/www/binbash.com.ar/public_html/index.html')
        sudo('rm -r /var/www/binbash.com.ar/www.binbash.com.ar/')
        sudo('echo "ServerName localhost" >> /etc/apache2/apache2.conf')
        sudo('a2ensite binbash.com.ar')
        sudo('chmod -R 755 /var/www')
        sudo('service apache2 restart')
```

```
sudo('cp /conf/apache2/ports.conf /etc/apache2/ports.conf')
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and
# README.Debian.gz
NameVirtualHost *:*
#Listen 80
#Listen 8080
<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    Listen 443
</IfModule>
<IfModule mod_gnutls.c>
```

```
    Listen 443
</IfModule>

sudo('cp /conf/apache2/binbash.com.ar.conf /etc/apache2/sites-
available/binbash.com.ar.conf')
NameVirtualHost 172.16.0.10:443
<VirtualHost 172.16.0.10:443>
    ServerAdmin exequielrafaela@gmail.com
    ServerName binbash.com.ar
    ServerAlias www.binbash.com.ar
    DocumentRoot /var/www/binbash.com.ar/public_html/
    ErrorLog /var/www/binbash.com.ar/logs/error.log
    CustomLog /var/www/binbash.com.ar/logs/access.log combined
    SSLEngine on
    SSLOptions +StrictRequire
    SSLCertificateFile /etc/ssl/certs/binbash.com.ar.crt
    SSLCertificateKeyFile /etc/ssl/private/binbash.com.ar.key
</VirtualHost>
```

Relevant Screenshots



Configuración servicio SQUID

4.- Configuración de servidor proxy HTTP

4.1.- El acceso al servicio de proxy será de manera explícita y será con autenticación.

4.1.1.- Usuario: nombre. Password: apellido

4.2.- Solo permitirá navegación a equipos se se encuentren en la misma red.

4.3.- Implementar y aplicar una regla que evite acceder a los siguientes sitios:

www.infobae.com/radio10/radio10-en-vivo.php

cadena3.com.ar/multimedia.asp

www.cadena3.com.ar/multimedia.asp

cadena3.com.ar/audios.asp

cadena3.com/multimedia.asp

cadena3.com/audios.asp

lv3.com.ar/multimedia.asp

lv3.com.ar/audios.asp

www.la100.com.ar/aspx/reproductor/reproductor.html

www.radiomitre.com.ar/popup_audio.asp

www.am970.com.ar/windowsmedia/lv2_envivo.htm

www.continental.com.ar/player.aspx

www.amdelplata.com/reproductor/reproductor2.php

www.amradioamerica.com/live.html

www.ambelgrano.com/site/programacion/radio.html

www.rivadavia.com.ar/streaming16/

mega.10musica.com/Radio

los40principales.com.ar/player/Radio/40Principales/index.html

www.fmrockandpop.com/index.php?option=com_content&view=article&id=114&Itemid=92

www.fmaspen.com/reproductor.htm

Instalación y configuración

La misma se realizó a través de una función con Python Fabric como se presenta a continuación:

```
def install_squid_ubuntu_14():
    """
Squid3 HTTP Proxy installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('=====================================', 'blue')
        print colored('INSTALLING : "Squid HTTP Proxy Server"', 'blue')
        print colored('=====================================', 'blue')
        sudo('apt-get install -y squid apache2-utils')
        sudo('cp /conf/squid/squid.conf /etc/squid3/squid.conf')
        sudo('cp /conf/squid/squid_passwd /etc/squid3/squid_passwd')
        sudo('cp /conf/squid/restricted-sites.squid /etc/squid3/restricted-
         sites.squid')
        sudo('service squid3 restart')
        sudo('cat /etc/squid3/squid.conf | egrep -v \'^#|^$\'')
        sudo('netstat -putona | grep 3128')
        sudo('cat /var/log/squid3/access.log')
```

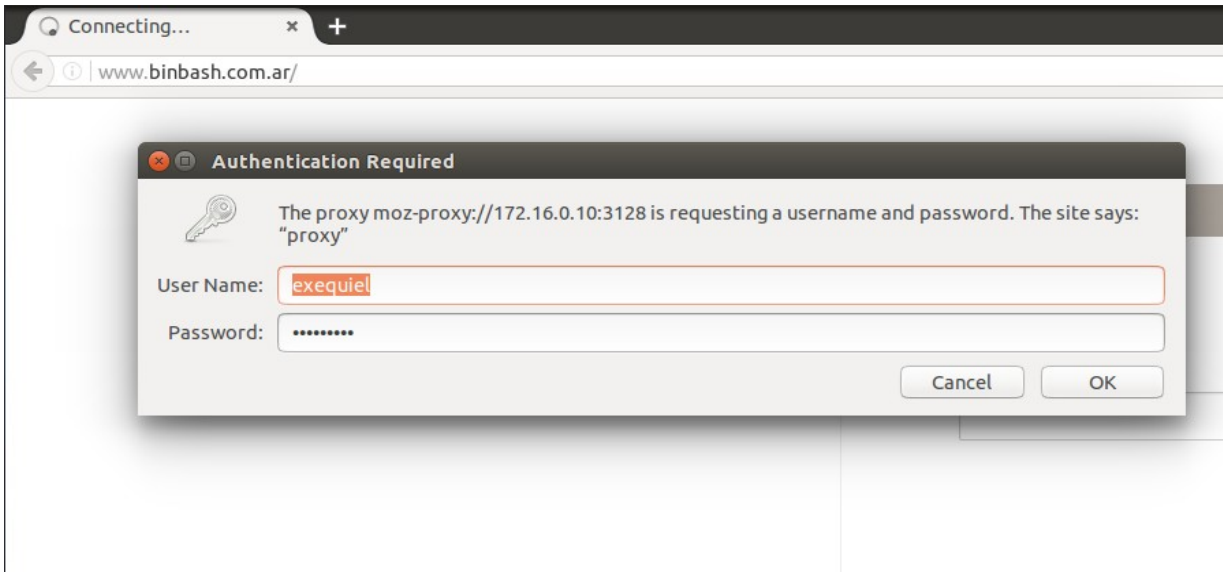
NOTE: Relevant configuration lines

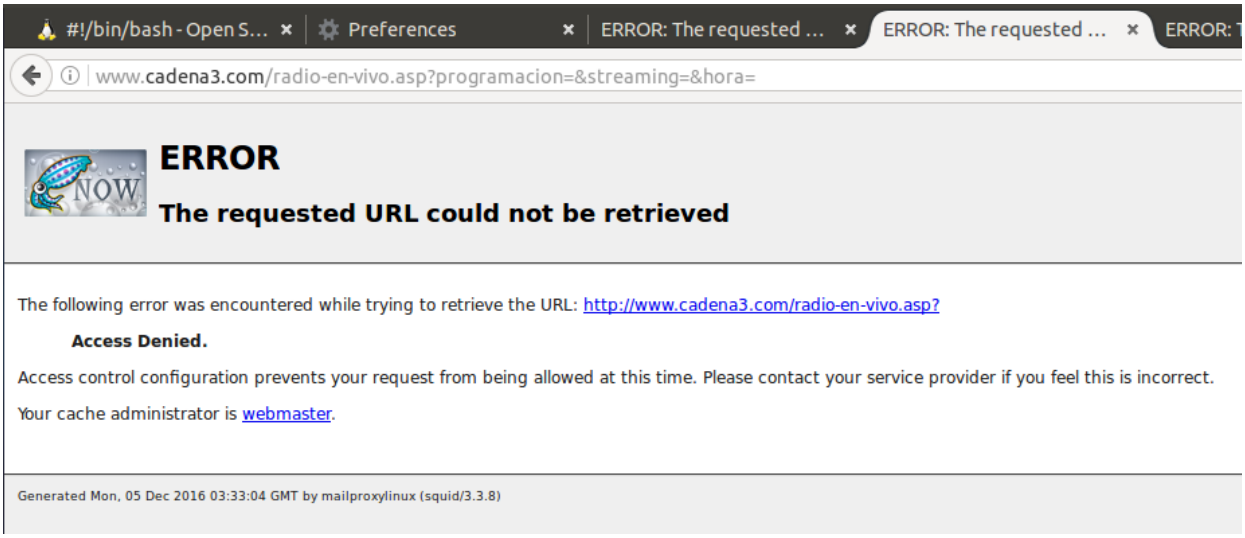
```
sudo('cat /etc/squid3/squid.conf | egrep -v \'^#|^$\'')

auth_param basic program /usr/lib/squid3/basic_ncsa_auth /etc/squid3/squid_passwd
auth_param basic realm proxy
acl localnet src 172.16.0.0/16   # RFC1918 possible internal network
acl localnet src 192.168.0.0/16  # RFC1918 possible internal network
acl ncsa_users proxy_auth REQUIRED
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
acl blocksites dstdomain "/etc/squid3/restricted-sites.squid"
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access deny blocksites
http_access allow ncsa_users
http_access allow localnet
http_access allow localhost
http_access deny all
http_port 172.16.0.10:3128
coredump_dir /var/spool/squid3
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
refresh_pattern .               0       20%     4320
```

```
sudo('cp /conf/squid/squid_passwd /etc/squid3/squid_passwd')
vagrant@mailproxylinux:/etc/squid3$ cat squid_passwd
user1:$apr1$0Z7k8Mwz$iz90g0cc2nr5Q58csQFd11
exequiel:$apr1$ApmiwmaE$jLXatxl6Z2R9y9gHThEfM1
```

```
sudo('cp /conf/squid/restricted-sites.squid /etc/squid3/restricted- sites.squid')
vagrant@mailproxylinux:/etc/squid3$ cat restricted-sites.squid
.infobae.com
.cadena3.com.ar
.cadena3.com
.lv3.com.ar
.la100.com.ar
.radiomitre.com.ar
.am970.com.ar
.continental.com.ar
.amdelplata.com
.amradioamerica.com
.ambelgrano.com
.rivadavia.com.ar
.mega.10musica.com
.los40principales.com.ar
.fmrockandpop.com
.fmaspen.com
```

Relevant Screenshots





Configuración servicio Postfix

3.- Configuración de servidor de correo

3.1.- Implementar un servidor de mail, utilizar el relayhost = 'tortuga.unc.edu.ar' y enviar un correo a ermirizio@gmail.com

3.2.- Crear un dominio de mail, con un correo electrónico asociado:

3.2.1.- Dominio 'apellido del alumno' (Se perfirió binbash.com.ar)

3.2.2.- nombre-del-alumno@apellido-del-alumno (luego exequiel@binbash.com.ar y barrirero@binbash.com.ar)

Instalación y configuración

La misma se realizó a través de una función con Python Fabric como se presenta a continuación:

```
def install_postfix_ubuntu_14():
    """
Postfix Internet Mailserver installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('================================', 'blue')
        print colored('INSTALLING : "Postfix Mailserver"', 'blue')
        print colored('================================', 'blue')
        sudo('DEBIAN_FRONTEND=noninteractive apt-get -y install postfix mailutils')
        sudo('cp /conf/postfix/main.cf /etc/postfix/main.cf')
        sudo('cp /conf/postfix/virtual /etc/postfix/virtual')
        sudo('postmap /etc/postfix/virtual')
        sudo('service postfix restart')
        sudo('netstat -putona | grep 25')
        sudo('cat /var/log/mail.log')


NOTE content of main.cf being copied when:
sudo('cp /conf/postfix/mani.cf /etc/postfix/main.cf')

# See /usr/share/postfix/main.cf.dist for a commented, more complete version
# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
# appending .domain is the MUA's job.
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h
readme_directory = no
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination
myhostname = mailproxylinux
#alias_maps = hash:/etc/aliases
virtual_alias_maps = hash:/etc/postfix/virtual
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
```

```
mydestination = binbash.com.ar, mailproxylinux, localhost.localdomain, localhost
relayhost = tortuga.unc.edu.ar
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 172.16.0.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

NOTE configuring domain binbash.com.ar emails:

```
sudo('cp /conf/postfix/virtual /etc/postfix/virtual')
sudo('postmap /etc/postfix/virtual')
vagrant@mailproxylinux:~$ cat /etc/postfix/virtual
exequiel@binbash.com.ar vagrant
barrirero@binbash.com.ar
```

CHECKING Postfix Service on oprt TCP #25:

```
vagrant@mailproxylinux:~$ netstat -putona | grep 25
(No info could be read for "-p": geteuid()=1000 but you should be root.)
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      -
off (0.00/0/0)
tcp6       0      0 :::25                   :::*                    LISTEN      -
off (0.00/0/0)
```

Servermail output y logs

```
vagrant@mailproxylinux:~$ echo "type *" | sudo mail
"/var/mail/root": 2 messages 2 new
>N   1 Exequiel Rafaela   Sun Dec  4 22:46  19/718   Mail local user
 N   2 Exequiel Rafaela   Sun Dec  4 22:50  19/677   Test
Return-Path: <exequielrafaela@gmail.com>
X-Original-To: root@binbash.com.ar
Delivered-To: root@binbash.com.ar
Received: from [172.16.0.1] (unknown [172.16.0.1])
        by mailproxylinux (Postfix) with ESMTP id 04A1D4089F
        for <root@binbash.com.ar>; Sun,  4 Dec 2016 22:46:22 +0000 (UTC)
To: root@binbash.com.ar
From: Exequiel Rafaela <exequielrafaela@gmail.com>
Subject: Mail local user
Message-ID: <3711b9d4-9848-4c9a-f71f-20b147128fa6@gmail.com>
Date: Sun, 4 Dec 2016 19:46:21 -0300
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
 Thunderbird/45.4.0
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 7bit
```

Mail for root user in binbash.com.ar domain

```
Return-Path: <exequielrafaela@noexiste.com>
X-Original-To: root@binbash.com.ar
Delivered-To: root@binbash.com.ar
Received: from [172.16.0.1] (unknown [172.16.0.1])
        by mailproxylinux (Postfix) with ESMTP id 98BE64089F
        for <root@binbash.com.ar>; Sun,  4 Dec 2016 22:50:45 +0000 (UTC)
To: root@binbash.com.ar
From: Exequiel Rafaela <exequielrafaela@noexiste.com>
Subject: Test
Message-ID: <a9dbd55e-f40d-6f51-8c16-a25a1e47b499@noexiste.com>
Date: Sun, 4 Dec 2016 19:50:45 -0300
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
 Thunderbird/45.4.0
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
```

```
Content-Transfer-Encoding: 7bit

Test

Saved 2 messages in /home/vagrant/mbox
Held 0 messages in /var/mail/root
```

NOTE Postfix logs summarized:
sudo('cat /var/log/mail.log')

```
Dec  4 22:28:31 vagrant-ubuntu-trusty-64 postfix/master[15541]: daemon started -- version
2.11.0, configuration /etc/postfix
Dec  4 22:46:21 vagrant-ubuntu-trusty-64 postfix/smtpd[15562]: connect from
unknown[172.16.0.1]
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/smtpd[15562]: 04A1D4089F:
client=unknown[172.16.0.1]
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/cleanup[15566]: 04A1D4089F: message-
id=<3711b9d4-9848-4c9a-f71f-20b147128fa6@gmail.com>
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 04A1D4089F:
from=<exequielrafaela@gmail.com>, size=624, nrcpt=1 (queue active)
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/smtpd[15562]: disconnect from
unknown[172.16.0.1]
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/local[15567]: 04A1D4089F:
to=<root@binbash.com.ar>, relay=local, delay=0.03, delays=0.03/0/0/0, dsn=2.0.0,
status=sent (
Dec  4 22:46:22 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 04A1D4089F: removed
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/smtpd[15689]: connect from
unknown[172.16.0.1]
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/smtpd[15689]: 98BE64089F:
client=unknown[172.16.0.1]
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/cleanup[15692]: 98BE64089F: message-
id=<a9dbd55e-f40d-6f51-8c16-a25a1e47b499@noexiste.com>
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 98BE64089F:
from=<exequielrafaela@noexiste.com>, size=580, nrcpt=1 (queue active)
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/smtpd[15689]: disconnect from
unknown[172.16.0.1]
...
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/smtp[15749]: 0D50D408A1:
to=<exequielrafaela@hotmail.com>, relay=tortuga.unc.edu.ar[200.16.16.23]:25, delay=0.35,
delaystus=bounced (host tortuga.unc.edu.ar[200.16.16.23] said: 554 5.7.1
<exequielrafaela@noexiste.com>: Sender address rejected: Access denied (in reply to RCPT
TO command))
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/cleanup[15748]: 6F051408A5: message-
id=<20161204231132.6F051408A5@mailproxylinux>
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/bounce[15750]: 0D50D408A1: sender non-
delivery notification: 6F051408A5
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 6F051408A5: from=<>,
size=2703, nrcpt=1 (queue active)
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 0D50D408A1: removed
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/smtp[15749]: 6F051408A5:
to=<exequielrafaela@noexiste.com>, relay=tortuga.unc.edu.ar[200.16.16.23]:25, delay=0.26,
delayounced (host tortuga.unc.edu.ar[200.16.16.23] said: 554 5.7.1 <>: Sender address
rejected: Access denied (in reply to RCPT TO command))
Dec  4 23:11:32 vagrant-ubuntu-trusty-64 postfix/qmgr[15545]: 6F051408A5:
removed22:50:45 vagrant-ubuntu-trusty-64 postfix/smtpd[15689]: connect from
unknown[172.16.0.1]
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/smtpd[15689]: 98BE64089F:
client=unknown[172.16.0.1]
Dec  4 22:50:45 vagrant-ubuntu-trusty-64 postfix/cleanup[15692]: 98BE64089F: message-
id=<a9dbd
```

Configuración servicio NFS

5.- Configuración de servidor NFS

5.1.- Compartir el directorio /srv/nfs a la red a la cual pertenece el servidor. Debe tener permiso de lectura y escritura. Tratar al usuario 'root' remoto como usuario 'nobody'

Instalación y configuración

```
def nfs_server_ubuntu(nfs_dir):
    """
Install in the host7s NFS Server under Debian/Ubuntu based systems
    :param nfs_dir: NFS directory to be shared
    """
    with settings(warn_only=False):
        sudo('apt-get update')
        sudo('apt-get -y install nfs-kernel-server')
        if exists( nfs_dir, use_sudo=True):
            print colored('##########################################', 'blue')
            print colored('####### Directory already created #########', 'blue')
            print colored('##########################################', 'blue')
        else:
            print colored('##########################################', 'red')
            print colored('###### Creating NFS share Directory #######', 'red')
            print colored('##########################################', 'red')
            sudo('mkdir ' + nfs_dir)
            sudo('chmod -R 777 '+nfs_dir+'/')
            sudo('chown nobody:nogroup ' + nfs_dir + '/')
        with settings(warn_only=True):
            #sudo('chkconfig nfs on')
            sudo('service rpcbind start')
            sudo('service nfs start')
        ip_addr = sudo('ifconfig eth1 | awk \'/inet /{print substr($2,6)}\'')
        netmask = sudo('ifconfig eth1 | awk \'/inet /{print substr($4,6)}\'')
        subnet_temp = iptools.ipv4.subnet2block(str(ip_addr) + '/' + str(netmask))
        subnet = subnet_temp[0]
        sudo('echo "' + nfs_dir + '     ' + subnet + '/' + netmask +
'(rw,sync,no_root_squash,no_subtree_check)" > /etc/exports')
        sudo('echo "' + nfs_dir + '     *(rw,sync,no_root_squash)" > /etc/exports')
        sudo('sudo exportfs -a')
        sudo('service nfs-kernel-server start')
```

Console Output:

```
==> mailproxylinux:                  ##########################################
==> mailproxylinux: [localhost] out: ####### Directory already created #########
==> mailproxylinux: [localhost] out: ##########################################
==> mailproxylinux: [localhost] out: [localhost] sudo: service rpcbind start
==> mailproxylinux: [localhost] out: start: Job is already running: rpcbind
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] sudo: service nfs start
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] sudo: ifconfig eth1 | awk '/inet /{print substr($2,6)}'
==> mailproxylinux: [localhost] out: 172.16.0.10
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] sudo: ifconfig eth1 | awk '/inet /{print substr($4,6)}'
==> mailproxylinux: [localhost] out: 255.255.255.0
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] sudo: echo "/srv/nfs
172.16.0.0/255.255.255.0(rw,sync,no_root_squash,no_subtree_check)" > /etc/exports
==> mailproxylinux: [localhost] out:
```

```
==> mailproxylinux: [localhost] sudo: echo "/srv/nfs     *(rw,sync,no_root_squash)" >
/etc/exports
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] sudo: sudo exportfs -a
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] out: exportfs: /etc/exports [1]: Neither 'subtree_check'
or 'no_subtree_check' specified for export "*:/srv/nfs".
==> mailproxylinux: [localhost] out: [localhost] out:   Assuming default behaviour
('no_subtree_check').
==> mailproxylinux: [localhost] out: [localhost] out:   NOTE: this default has changed
since nfs-utils version 1.0.x
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] out: [localhost] sudo: service nfs-kernel-server start
==> mailproxylinux: [localhost] out:
==> mailproxylinux: [localhost] out:  * Exporting directories for NFS kernel daemon...
==> mailproxylinux: [localhost] out: [localhost] out: exportfs: /etc/exports [1]: Neither
'subtree_check' or 'no_subtree_check' specified for export "*:/srv/nfs".
==> mailproxylinux: [localhost] out: [localhost] out:   Assuming default behaviour
('no_subtree_check').
==> mailproxylinux: [localhost] out: [localhost] out:   NOTE: this default has changed
since nfs-utils version 1.0.x
==> mailproxylinux: [localhost] out: [localhost] out:
==> mailproxylinux: [localhost] out: [localhost] out:    ...done.
==> mailproxylinux: [localhost] out: [localhost] out:  * Starting NFS kernel daemon
==> mailproxylinux: [localhost] out: [localhost] out:    ...done.
==> mailproxylinux: [localhost] out: [localhost] out:
==> mailproxylinux: [localhost] out: Done.
==> mailproxylinux: [localhost] out:
==> mailproxylinux: Disconnecting from localhost... done.
==> mailproxylinux: [localhost] out:
```

Configuración servicio Logrotate

8.- Configuración de logrotate

8.1.- Configurar la rotación de logs para que no superen los 30 días de almacenamiento. Todo log se debe rotar una vez por día.

El resto de las opciones dejar por defecto. Esto aplica a los logs de los siguientes servicios:

- Apache

- Squid

- Postfix

Instalación y configuración

```
def install_logrotate_ubuntu_14():
    """
Squid3 HTTP Proxy installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('====================================', 'blue')
        print colored('INSTALLING : "Logrotate Service      "', 'blue')
        print colored('====================================', 'blue')
        sudo('apt-get install -y logrotate')
        sudo('cp /conf/logrotate/logrotate.conf /etc/logrotate/logrotate.conf')
        sudo('cp /conf/logrotate/squid3 /etc/logrotate/lograte.d/squid3')
        sudo('cp /conf/logrotate/apache2 /etc/logrotate/lograte.d/apache2')
        sudo('cp /conf/logrotate/postfix /etc/logrotate/lograte.d/postfix')

NOTE Logrotate config file to achieve de instructions 8.1 :
sudo('cp /conf/logrotate/logrotate.conf /etc/logrotate/logrotate.conf')
# see "man logrotate" for details


# rotate log files weekly
#monthly
#weekly
daily
# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}
# system-specific logs may be configured here
```

```
sudo('cp /conf/logrotate/squid3 /etc/logrotate/lograte.d/squid3')

#
#       Logrotate fragment for squid3.
#
/var/log/squid3/*.log {
         daily
         compress
         delaycompress
         rotate 2
         missingok
         nocreate
         sharedscripts
         prerotate
                 test ! -x /usr/sbin/sarg-reports || /usr/sbin/sarg-reports
         endscript
         postrotate
                 test ! -e /var/run/squid3.pid || test ! -x /usr/sbin/squid3 ||
/usr/sbin/squid3 -k rotate
         endscript
}


sudo('cp /conf/logrotate/apache2 /etc/logrotate/lograte.d/apache2')
var/log/apache2/*.log {

         weekly
         missingok
         rotate 52
         compress
         delaycompress
         notifempty
         create 640 root adm
         sharedscripts
         postrotate
                 if /etc/init.d/apache2 status > /dev/null ; then \
                     /etc/init.d/apache2 reload > /dev/null; \
                 fi;
         endscript
         prerotate
                 if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                          run-parts /etc/logrotate.d/httpd-prerotate; \
                 fi; \
         endscript
}



sudo('cp /conf/logrotate/postfix /etc/logrotate/lograte.d/postfix')

#

#       Logrotate fragment for postfix.
#
/var/log/mail.log {
         daily
         compress
         delaycompress
         rotate 2
         missingok
         create 0644 postfix postfix
}
```

Console Output:

```
vagrant@mailproxylinux:/etc/cron.daily$ cat logrotate
#!/bin/sh

# Clean non existent log file entries from status file
cd /var/lib/logrotate
test -e status || touch status
head -1 status > status.clean
sed 's/"//g' status | while read logfile date
do
    [ -e "$logfile" ] && echo "\"$logfile\" $date"
done >> status.clean
mv status.clean status

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

Configuración monitoreo Munin

1.- Instalación de servicio munin.

1.1.- Monitorear localhost.

1.2.- Asegurar que se encuentren los siguientes servicios monitoreados

1.2.1.- munin, apache*, postfix*, firewall*, nfs*, squid*

Instalación y configuración

La misma se realizó a través de una función con Python Fabric como se presenta a continuación:

```
def install_munin_master_ubuntu_14():
    """
Munin Master HTTP Monitoring installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('=======================================', 'blue')
        print colored('INSTALLING : "Munin Monitoring Service"', 'blue')
        print colored('=======================================', 'blue')
        sudo('apt-get install -y apache2 apache2-utils libcgi-fast-perl libwww-perl
            libapache2-mod-fcgid munin')
        sudo('apt-get install munin-plugins-extra')
        with settings(warn_only=True):
            sudo('a2enmod fcgid')
            sudo('cp /conf/munin/munin.conf /etc/munin/munin.conf')
            sudo('cp /conf/munin/apache.conf /etc/munin/apache.conf')
            sudo('cp /conf/munin/apache.conf /etc/munin/plugin-conf.d/munin-node')
        #Activating extra plugins (Apache & Squid)
        with settings(warn_only=True):
            sudo('/usr/sbin/munin-node-configure --suggest')
            sudo('/usr/sbin/munin-node-configure --shell | sudo sh')
            sudo('ln -s /usr/share/munin/plugins/squid_cache /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_icp /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_objectsize
                 /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_requests /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_traffic /etc/munin/plugins/')
            sudo('/usr/sbin/munin-node-configure --suggest | egrep \'squid|apache|nfs|
                  postfix|firewall|munin\'')
        #Restarting services
        sudo('service apache2 restart')
        sudo('service munin-node restart')


def install_munin_node_ubuntu_14():
    """
Munin Node HTTP Monitoring installation in Ubuntu 14.04.
    """
    with settings(warn_only=False):
        print colored('=======================================', 'blue')
        print colored('INSTALLING : "Munin Monitoring Service"', 'blue')
        print colored('=======================================', 'blue')
        sudo('apt-get install -y munin-node libwww-perl')
        sudo('apt-get install munin-plugins-extra')
        sudo('cp /conf/munin/munin-node.conf /etc/munin/munin-node.conf')
```

```
        # Activating extra plugins (Apache & Squid)
        with settings(warn_only=True):
            sudo('/usr/sbin/munin-node-configure --suggest')
            sudo('/usr/sbin/munin-node-configure --shell | sudo sh')
            sudo('ln -s /usr/share/munin/plugins/squid_cache /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_icp /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_objectsize
                 /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_requests /etc/munin/plugins/')
            sudo('ln -s /usr/share/munin/plugins/squid_traffic /etc/munin/plugins/')
            sudo('/usr/sbin/munin-node-configure --suggest | egrep \'squid|apache|nfs|
                  postfix|firewall|munin\'')
        #  Restarting services
        sudo('service munin-node restart')
```

NOTE: Relevant configuration lines (MASTER/Server SIDE)

```
sudo('cp /conf/munin/munin.conf /etc/munin/munin.conf')
# Example configuration file for Munin, generated by 'make build'
# The next three variables specifies where the location of the RRD
# databases, the HTML output, logs and the lock/pid files.  They all
# must be writable by the user running munin-cron.  They are all
# defaulted to the values you see here.
#
dbdir  /var/lib/munin
htmldir /var/cache/munin/www
logdir /var/log/munin
rundir  /var/run/munin

# a simple host tree
#[localhost.localdomain]
[MuninMaster]
    address 127.0.0.1
    use_node_name yes
[MuninNode]
    address 172.16.0.50
    use_node_name yes
```

NOTE: Munin APACHE CONF

```
sudo('cp /conf/munin/apache.conf /etc/munin/apache.conf')
# Enable this for template generation
Alias /munin /var/cache/munin/www
<Directory /var/cache/munin/www>
        Require all granted
        Options FollowSymLinks SymLinksIfOwnerMatch

    <IfModule mod_expires.c>
        ExpiresActive On
        ExpiresDefault M310
    </IfModule>
</Directory>

ScriptAlias /munin-cgi/munin-cgi-graph /usr/lib/munin/cgi/munin-cgi-graph
<Location /munin-cgi/munin-cgi-graph>
        #Order allow,deny
        #Allow from localhost 127.0.0.0/8 ::1
        Require all granted
        Options FollowSymLinks SymLinksIfOwnerMatch
        # AuthUserFile /etc/munin/munin-htpasswd
        # AuthName "Munin"
        # AuthType Basic
        # require valid-user
        <IfModule mod_fcgid.c>
```

```
            SetHandler fcgid-script
        </IfModule>
        <IfModule !mod_fcgid.c>
            SetHandler cgi-script
        </IfModule>
</Location>
ScriptAlias /munin-cgi/munin-cgi-html /usr/lib/munin/cgi/munin-cgi-html
<Location /munin-cgi/munin-cgi-html>
        #Order allow,deny
        #Allow from localhost 127.0.0.0/8 ::1
        Require all granted
        Options FollowSymLinks SymLinksIfOwnerMatch
        # AuthUserFile /etc/munin/munin-htpasswd
        # AuthName "Munin"
        # AuthType Basic
        # require valid-user
        <IfModule mod_fcgid.c>
            SetHandler fcgid-script
        </IfModule>
        <IfModule !mod_fcgid.c>
            SetHandler cgi-script
        </IfModule>
</Location>


NOTE: Munin extra plugin configs
sudo('cp /conf/munin/apache.conf /etc/munin/plugin-conf.d/munin-node')
[courier_mta_mailvolume]
group adm
[cps*]
user root
[df*]
env.warning 92
env.critical 98
[exim_mailqueue]
group adm, (Debian-exim)
[exim_mailstats]
group adm, (Debian-exim)
env.logdir /var/log/exim4/
env.logname mainlog
[fw_conntrack]
user root
[fw_forwarded_local]
user root
[hddtemp_smartctl]
user root
[hddtemp2]
user root
[if_*]
user root
[if_err_*]
user nobody
[ip_*]
user root
[ipmi_*]
user root
[mysql*]
user root
env.mysqlopts --defaults-file=/etc/mysql/debian.cnf
env.mysqluser debian-sys-maint
env.mysqlconnection DBI:mysql:mysql;mysql_read_default_file=/etc/mysql/debian.cnf
[postfix_mailqueue]
user postfix
[postfix_mailstats]
group adm
[postfix_mailvolume]
```

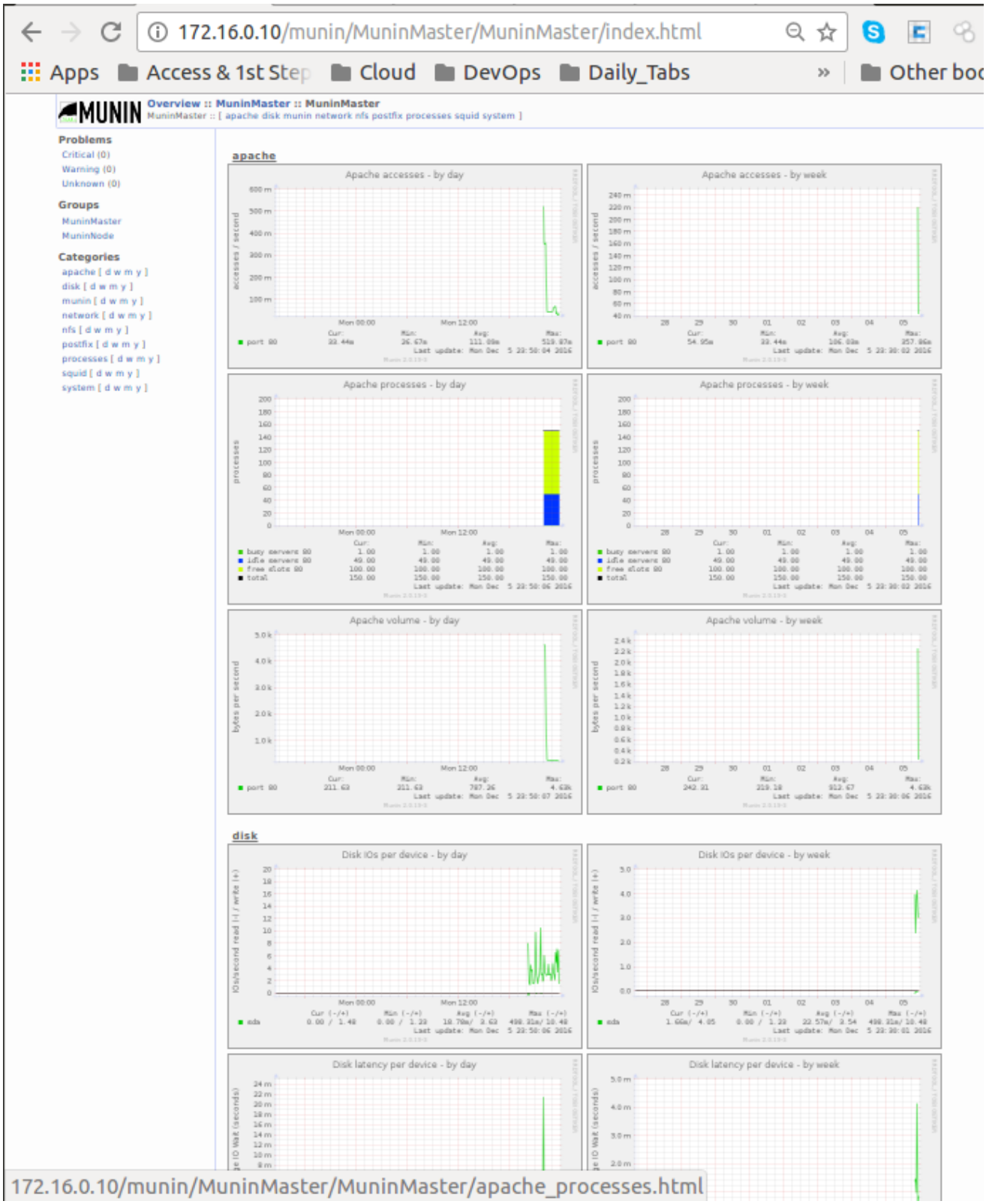
```
group adm
env.logfile mail.log
[sendmail_*]
user smmta
[smart_*]
user root
[vlan*]
user root
[ejabberd*]
user ejabberd
env.statuses available away chat xa
env.days 1 7 30
[dhcpd3]
user root
env.leasefile /var/lib/dhcp3/dhcpd.leases
env.configfile /etc/dhcp3/dhcpd.conf
[jmx_*]
env.ip 127.0.0.1
env.port 5400
[samba]
user root
[munin_stats]
user munin
group munin
[postgres_*]
user postgres
env.PGUSER postgres
env.PGPORT 5432
[fail2ban]
user root
[squid*]
env.squiduser exequiel
env.squidpasswd barrirero
env.squidhost 127.0.0.1
env.squidport 3128
```

NOTE: Relevant configuration lines (NODE SIDE)

```
sudo('cp /conf/munin/munin-node.conf /etc/munin/munin-node.conf')
#
# Example config-file for munin-node
#
log_level 4
log_file /var/log/munin/munin-node.log
pid_file /var/run/munin/munin-node.pid
background 1
setsid 1
user root
group root
# This is the timeout for the whole transaction.
# Units are in sec. Default is 15 min
#
# global_timeout 900
# This is the timeout for each plugin.
# Units are in sec. Default is 1 min
#
# timeout 60
# Regexps for files to ignore
ignore_file [\#~]$
ignore_file DEADJOE$
ignore_file \.bak$
ignore_file %$
ignore_file \.dpkg-(tmp|new|old|dist)$
ignore_file \.rpm(save|new)$
ignore_file \.pod$
# Set this if the client doesn't report the correct hostname when
# telnetting to localhost, port 4949
#
```

```
#host_name localhost.localdomain
# A list of addresses that are allowed to connect.  This must be a
# regular expression, since Net::Server does not understand CIDR-style
# network notation unless the perl module Net::CIDR is installed.  You
# may repeat the allow line as many times as you'd like
allow ^172\.16\.0\.10$
allow ^127\.0\.0\.1$
allow ^::1$

# Which address to bind to;
host 172.16.0.50
# host 127.0.0.1
# And which port
port 4949
```

Relevant Screenshots


172.16.0.10/munin/MuninMaster/MuninMaster/index.html
Apps
Access & 1st Step
Cloud
DevOps
Daily_Tabs
Other boo
MUNIN
Overview :: MuninMaster :: MuninMaster
MuninMaster :: [apache disk munin network nfs postfix processes squid system]
Problems
Critical (0)
Warning (0)
Unknown (0)
Groups
MuninMaster
MuninNode
Categories
apache [d w m y]
disk [d w m y]
munin [d w m y]
network [d w m y]
nfs [d w m y]
postfix [d w m y]
processes [d w m y]
squid [d w m y]
system [d w m y]
apache
Apache accesses - by day
Apache accesses - by week
Apache processes - by day
Apache processes - by week
Apache volume - by day
Apache volume - by week
disk
Disk IOs per device - by day
Disk IOs per device - by week
Disk latency per device - by day
Disk latency per device - by week
172.16.0.10/munin/MuninMaster/MuninMaster/apache_processes.html

Configuración FIREWALL (IPtables)

2.- Configuración de firewall

2.1- Configurar el firewall para que el servidor pueda realizar cualquier conexión saliente, pero solo pueda recibir conexiones para los servicios configurados.

Solo se deben aceptar conexiones de la red a la cual pertenece el servidor, es decir 172.16.0.0/24 (se agrega tambien la red host de la VM – 192.168.0.0/24).

Instalación y configuración

La misma se realizó a través de una función con Python Fabric como se presenta a continuación:

```
print colored('############################', 'blue')
print colored('##### START FIREWALL #####', 'blue')
print colored('############################', 'blue')
# CONSIDER:
# root@mailproxylinux:/etc/munin/plugin-conf.d# netstat -putona | egrep -i '22|3128|25|
80|443|4949'
# tcp        0      0 0.0.0.0:52526           0.0.0.0:*               LISTEN
781/rpc.statd    off (0.00/0/0)
# tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
1862/sshd        off (0.00/0/0)
# tcp        0      0 172.16.0.10:3128        0.0.0.0:*               LISTEN
22317/squid3     off (0.00/0/0)
# tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
22172/master     off (0.00/0/0)
# tcp        0      0 10.0.2.15:22            10.0.2.2:50106          ESTABLISHED
3855/sshd: vagrant  keepalive (2683.93/0/0)
# tcp        0      0 10.0.2.15:22            10.0.2.2:40326          ESTABLISHED
24805/sshd: vagrant keepalive (3699.74/0/0)
# tcp        0      0 10.0.2.15:22            10.0.2.2:40600          ESTABLISHED
27315/sshd: vagrant keepalive (2094.11/0/0)
# tcp6       0      0 :::80                   :::*                    LISTEN
23617/apache2    off (0.00/0/0)
# tcp6       0      0 :::4949                 :::*                    LISTEN
23695/perl       off (0.00/0/0)
# tcp6       0      0 :::22                   :::*                    LISTEN
1862/sshd        off (0.00/0/0)
# tcp6       0      0 :::25                   :::*                    LISTEN
22172/master     off (0.00/0/0)
# tcp6       0      0 :::443                  :::*                    LISTEN
23617/apache2    off (0.00/0/0)
# udp        0      0 0.0.0.0:51013           0.0.0.0:*
22317/squid3     off (0.00/0/0)
# udp        0      0 0.0.0.0:52293           0.0.0.0:*
781/rpc.statd    off (0.00/0/0)
# udp6       0      0 ::1:46032               ::1:40978               ESTABLISHED
22317/squid3     off (0.00/0/0)
# udp6       0      0 ::1:40978               ::1:46032               ESTABLISHED
22320/(pinger)   off (0.00/0/0)
# udp6       0      0 :::54445                :::*
22317/squid3     off (0.00/0/0)
# To stop Ipv4 based iptables firewall
sudo('iptables-save > $HOME/firewall.txt')
sudo('iptables -X')
sudo('iptables -t nat -F')
sudo('iptables -t nat -X')
sudo('iptables -t mangle -F')
sudo('iptables -t mangle -X')
sudo('iptables -P INPUT ACCEPT')
sudo('iptables -P FORWARD ACCEPT')
sudo('iptables -P OUTPUT ACCEPT')
# To stop Ipv6 based iptables firewall, enter:
```

```
sudo('ip6tables-save > $HOME/firewall-6.txt')
sudo('ip6tables -X')
sudo('ip6tables -t mangle -F')
sudo('ip6tables -t mangle -X')
sudo('ip6tables -P INPUT ACCEPT')
sudo('ip6tables -P FORWARD ACCEPT')
sudo('ip6tables -P OUTPUT ACCEPT')
# To start Ipv4 based iptables firewall, enter:
# SSH management
sudo('iptables -A INPUT -p tcp -s 10.0.2.0/24 --dport ssh -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport ssh -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport ssh -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport ssh -j DROP')
# SMTP POSTFIX
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport 25 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 25 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 25 -j DROP')
# HTTP APACHE2
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport 80 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 80 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 80 -j DROP')
# HTTPS APACHE2
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport 443 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 443 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 443 -j DROP')
# PROXY SQUID3
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport 3128 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 3128 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 3128 -j DROP')
# PROXY SQUID3
sudo('iptables -A INPUT -p tcp -s 172.16.0.0/24 --dport 4949 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 4949 -j ACCEPT')
sudo('iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 4949 -j DROP')
# DROP EVERY OTHER TRAFFIC
sudo('iptables -A INPUT -j DROP')


==> mailproxylinux: ######################################
==> mailproxylinux: END FIREWALL - FILTER TABLE STATUS:
==> mailproxylinux: ######################################
==> mailproxylinux: [localhost] sudo: iptables -L
==> mailproxylinux: Chain INPUT (policy ACCEPT)
==> mailproxylinux: target     prot opt source               destination
==> mailproxylinux: ACCEPT     tcp  --  10.0.2.0/24         anywhere            tcp
dpt:ssh
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24       anywhere            tcp
dpt:ssh
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24      anywhere            tcp
dpt:ssh
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24       anywhere            tcp
dpt:smtp
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24      anywhere            tcp
dpt:smtp
==> mailproxylinux: DROP       tcp  --  anywhere            anywhere            tcp
dpt:smtp
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24       anywhere            tcp
dpt:http
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24      anywhere            tcp
dpt:http
==> mailproxylinux: DROP       tcp  --  anywhere            anywhere            tcp
dpt:http
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24       anywhere            tcp
dpt:https
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24      anywhere            tcp
dpt:https
==> mailproxylinux: DROP       tcp  --  anywhere            anywhere            tcp
dpt:https
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24       anywhere            tcp
```

```
dpt:3128
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24       anywhere             tcp
dpt:3128
==> mailproxylinux: DROP       tcp  --  anywhere             anywhere             tcp
dpt:3128
==> mailproxylinux: ACCEPT     tcp  --  172.16.0.0/24        anywhere             tcp
dpt:munin
==> mailproxylinux: ACCEPT     tcp  --  192.168.0.0/24       anywhere             tcp
dpt:munin
==> mailproxylinux: DROP       tcp  --  anywhere             anywhere             tcp
dpt:munin
==> mailproxylinux:
==> mailproxylinux: Chain FORWARD (policy ACCEPT)
==> mailproxylinux: target     prot opt source               destination
==> mailproxylinux:
==> mailproxylinux: Chain OUTPUT (policy ACCEPT)
==> mailproxylinux: target     prot opt source               destination
==> mailproxylinux: ############################
==> mailproxylinux: ## NETWORK CONFIGURATION #
==> mailproxylinux: ############################
==> mailproxylinux: [localhost] sudo: ip addr show
==> mailproxylinux:
==> mailproxylinux: 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default
==> mailproxylinux:     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
==> mailproxylinux:     inet 127.0.0.1/8 scope host lo
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 ::1/128 scope host
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux: 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
==> mailproxylinux:     link/ether 08:00:27:48:af:80 brd ff:ff:ff:ff:ff:ff
==> mailproxylinux:     inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 fe80::a00:27ff:fe48:af80/64 scope link
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux: 3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
==> mailproxylinux:     link/ether 08:00:27:56:c1:74 brd ff:ff:ff:ff:ff:ff
==> mailproxylinux:     inet 172.16.0.10/24 brd 172.16.0.255 scope global eth1
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:     inet6 fe80::a00:27ff:fe56:c174/64 scope link
==> mailproxylinux:        valid_lft forever preferred_lft forever
==> mailproxylinux:
==> mailproxylinux: Done.
```