REDFONE communications

Trixbox High-Availability with foneBRIDGE Tutorial

VoIPon www.voipon.co.uk sales@voipon.co.uk Tel: +44 (0)1245 808195 Fax: +44 (0)1245 600030

Table of Contents

Introduction 1

1.1 Overview

This tutorial aims to cover all the important elements for deploying a dual Trixbox server cluster with the goal of achieving fault tolerance and high availability. This guide will not cover the intricacies and nuances of setting up dial plans or SIP stations. The same tools and concepts could very easily be employed on any standard Asterisk install.

In our Tutorial we will cover a medium to large sized Trixbox deployment (25-100 users) that consists of one channel bank for analog extensions and SIP phones for the remaining telephone stations. A T1 PRI provides our primary incoming and outbound path.

1.1.1 Core components & requirements

*2 server grade platforms with dual Ethernet capability.

*Standard Trixbox 2.0 install (may work on earlier releases)

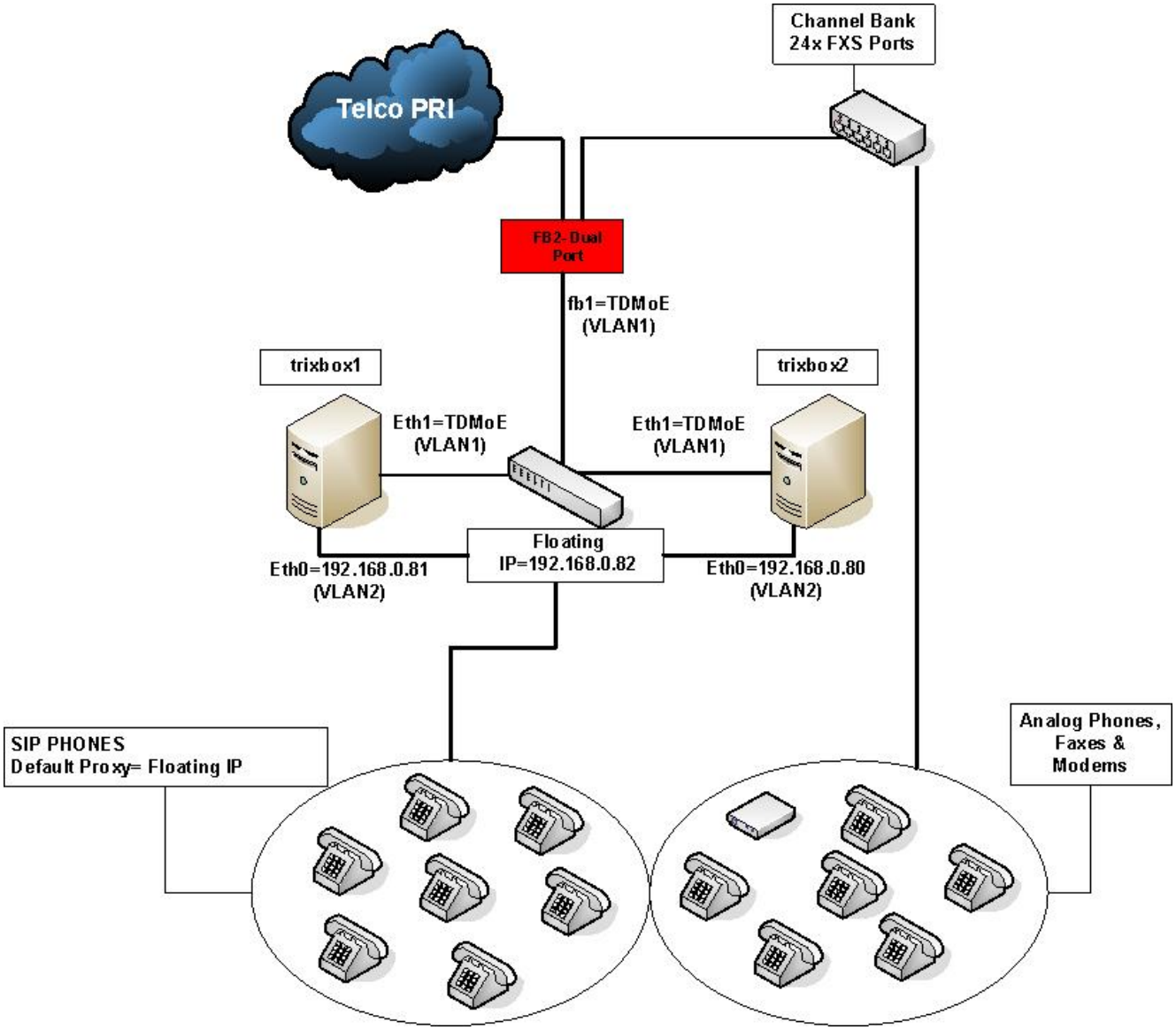
*Managed Layer 2 Switch with separate VLANs for foneBRIDGE2 and IP traffic

*foneBRIDGE2 for T1/E1 connectivity and built-in failover capability

*Heartbeat software package from Linux-HA

*Supported zaptel source code for the foneBRIDGE2

See diagram below for detailed illustration;



1.1.2 Operational Overview

In our scenario both Trixbox servers will be installed and configured to be a mirror image of one another. They will be deployed in an active/passive mode where by the primary server (trixbox1) is responsible for handling all TDM, SIP and Analog communication until it experiences a hardware fault. In the event of a fault on the primary server, the secondary (trixbox2) will take over control of our PBX deployment until the primary can be serviced and returned to operation. This functionality is handled by the 'heartbeat' application running on both Trixbox servers.

Heartbeat sends heartbeat packets across the network and/or serial ports to the other instances of Heartbeat as a sort of keep-alive message. When heartbeat packets are no longer received, the node is assumed to be dead, and any services (resources) it was providing are failed over to the other node.

On the SIP side of our implementation we will leverage the 'floating IP' (192.168.0.82) controlled by heartbeat to registrar our SIP clients against. If our primary server experiences a fault, our floating IP address will be quickly re-assigned to the secondary server. This allows our SIP clients to maintain registration without the need to reconfigure the clients to register to a new server.

On the TDM side (T1, Channel Bank) our connectivity is controlled by the fonebridge T1/E1 to TDM over Ethernet (TDMoE) appliance. The fonebridge is configured and controlled by a small executable script (fonulator) installed on both Trixbox servers. Again, in the event of a fault on trixbox1, heartbeat will notify trixbox2 to take over communication. At this point through the heartbeat 'resources', the fonulator utility on the trixbox2 server will be executed sending a small update packet to the fonebridge, quickly reconfiguring the device to communicate with our back-up server. Along with the execution of fonulator, heartbeat will also execute the start-up script for Asterisk fully bringing trixbox2 on-line. To ensure quick fail-over both trixbox servers will have zaptel loaded. During failover only the fonulator and asterisk will be started.

High Availability Install 2

2.1 Heartbeat Install & Configuration

All following configs are to be completed on both Trixbox servers and should be identical unless otherwise noted.

After a normal Trixbox 2.0 install;

1. Login to your root account with the password that you entered in the setup.

2. Type 'yum install update' and press enter. Enter 'y' when it asks you if this is ok.

3. Once that is completed type 'yum install heartbeat' and press enter. Do the same as above in the prompt (enter y)

4. Only three files need to be configured for heartbeat; ***ha.cf***, ***haresources*** *and* ***authkeys***

5. Edit '/etc/ha.d/ha.cf' with your favorite editor (vi,nano, WinSCP, etc.)

6. The following is the only information required for a basic setup and is the configuration based on the diagram:

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
keepalive 2
deadtime 10
bcast eth1
node trixbox1 trixbox2
auto_failback off
```

7. Save changes and exit

8. Edit '/etc/ha.d/haresources'and add;

```
trixbox1 192.168.0.82 fonulator asterisk
```

haresources defined: 'hostname of primary server' 'floating IP Address' 'names of services/scripts to be executed during failover'.

9. Edit '/etc/ha.d/authkeys' and add;

```
auth1
1 sha1 SuPerS&cretP@$$werd (change password to your own)
```

10. Change permissions on authkeys to 600

```
#chmod 600 authkeys
```

11. Ensure heartbeat starts up properly

```
#service heartbeat start
```

If you encounter any issues or error messages during startup, check your logs while starting up the service with;

```
# tail -f /var/log/ha-debug
```

2.1.1 IP and Hostname Resolution

Since heartbeat depends heavily on reliable IP communication it is important to ensure that IP address to hostname resolution works. Add the following to your /etc/hosts;

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1                localhost
127.0.0.1                asterisk1.local
192.168.0.81             trixbox2
192.168.0.80             trixbox1
```

Validate hostname-to-IP Address resolution with the ping command;

```
#ping trixbox1
```

```
#ping trixbox2
```

2.1.2 Start-Up Scripts Installation

1. Change to your /etc/init.d/ directory
   #cd /etc/init.d/
2. Download the fonulator startup script template;
   #wget http://support.red-fone.com/downloads/fonulator/fonulator_initd_script
3. Change name of startup script to ‘fonulator
   #mv fonulator_init_script fonulator
4. Ensure script has execution properties
   #chmod a+x fonulator
5. Backup existing zaptel start-up script
   #mv zaptel zaptel.bak
6. Download Redfone zaptel startup script
   #wget http://support.red-fone.com/downloads/zaptel/start_script
7. Change name of start up script
   #mv start_script zaptel
8. Ensure script has execution properties and is enabled for start-up during boot.
   #chmod a+x zaptel
   #chkconfig --level 3 zaptel on

FoneBRIDGE2 Install

3

3.1 Zaptel Install

Before installation, you need to download the patched verison of zaptel which is located at http://support.red-fone.com/downloads/zaptel/

1. Move to the /usr/src directory and download the pre-patched version of zaptel.

#cd /usr/src

#wget http://support.red-fone.com/downloads/zaptel/zaptel-1.2.11.tar.gz

2. Untar zaptel source code

    #tar –xvzf zaptel-1.2.11.tar.gz

3. Move to zaptel directory

    #cd zaptel-1.2.11

4. Install the correct kernel source for your kernel version

    #yum install kernel-devel

5. To correct a known zaptel build issue in CentOS/Trixbox enter the following;

    #sed -i s/rw_lock/rwlock/ /usr/src/kernels/`uname -r`-`uname -m`/include/linux/spinlock.h

6. Build zaptel

    #make clean; make; make install

7. Download and install fonulator, the fonebridge configuration utility.

```
#cd /usr/local/sbin

#wget http://support.red-fone.com/downloads/fonulator/fonulator32-2.6

#mv fonulator32-2.6 fonulator

#chmod a+x fonulator
```

3.1.1 Configure Zaptel

Open /etc/zaptel.conf with your favorite editor and add the following lines. Please note that these configs apply only to this documented scenario and are by no means applicable to all fonebridge install scenarios. No other parameters are necessary for this particular install.

```
# zaptel.conf config file
#
dynamic=eth,eth1/00:50:C2:65:D0:7E/0,24,2
dynamic=eth,eth1/00:50:C2:65:D0:7E/1,24,1
#
# PRI Channels
bchan=1-23
dchan=24
#
# Channel Bank Lines
fxoks=25-48
#
loadzone=us
defaultzone=us
```

3.1.2 Configure Zapata.conf

As a simple illustration and for reference our zapata.conf configuration would be as follows for this scenario. Your configuration may be different.

```
[channels]
immediate=no
callgroup=1
switchtype=national
signalling=pri_cpe
context=default
resetinterval=never
group=1
; T1 PRI
channel => 1-23
; Channel Bank
group=2
signalling=fxo_ks
channel => 25-48
```

3.1.3 Configure redfone.conf

The /etc/redfone.conf file is called by the fonulator utility and sends all config information to the fonebridge over Ethernet, communicating on a MAC Address-to-MAC Address basis.

```
[globals]
fb1=00:50:C2:65:D0:2C
fb2=00:50:C2:65:D0:2D
server1=00:16:36:76:09:91
card=eth1,fb1

[span1]
span=1,0,0,esf,b8zs
server1
fb1
pri

[span2]
span=2,0,0,esf,b8zs
server1
fb1
```

In the ‘globals’ field we establish our basic information pertaining to the MAC addresses in our implementation. fb1 and fb2 are the unique MAC addresses assigned to our fonebridge. The ‘server1=’ address is the MAC address of our Trixbox server.

NOTE: The only difference in the configurations of the redfone.conf file on both Trixbox servers is the ‘serverN=MACaddress’ since each server should have a unique MAC address assigned to its Ethernet interface.

The ‘card=’ parameter specifies that we will be sending the update/configure packet out eth1 on our Trixbox server, destined to fb1 on the fonebridge.

Under the ‘spanX’ parameters we specify port number, line encoding and framing. The ‘server1’specifies that the fonebridge will be communicating with the MAC address of server1, defined above under the Globals, and all traffic from the fonebridge side will be transmitted via fb1. Finally the ‘pri’ parameter indicates that this span or port is connected to a T1 PRI with 23 B + 1 D channels. The absence of the ‘pri’ parameter places this span or port in CAS/RBS mode which is used by Channel Banks and other telco equipment that requires CAS/RBS signaling.

Appendices

4

4.1 Testing & Troubleshooting Heartbeat

The simplest method for verifying heartbeat is established and communicating between your two Trixbox nodes is to start the service on each server.

#/etc/init.d/heartbeat start
or
#service heartbeat start

If for some reason you receive an error message when starting heartbeat on a certain node it's always helpful to check your logs as they will usually give you a good indication as to what is wrong.

Try opening up another shell session and actively monitoring (tail –f) the logs while starting heartbeat.

tail -f /var/log/ha-debug
and/or
#tail –f /var/log/ha-log

To verify the failover process is occurring properly, try stopping heartbeat on your primary server (trixbox1) and monitoring the ha-debug and ha-log on your secondary (trixbox2) to see if take over is successful.

The floating IP address is normally installed as an alias or pseudo IP (eth0:0) attached to your primary IP address. Ex.;

```
# ifconfig eth0:0
eth0      Link encap:Ethernet  HWaddr 00:0D:61:7E:2E:2D
      inet addr:192.168.0.82  Bcast:192.168.0.255  Mask:255.255.255.0
```

4.1.1 References