High Availability Setup Guide

Revised: 2008-02-05

Table of Contents

Introduction

This document describes the extra steps needed to configure a pair of systems for redundant operation to provide highly available calling service. It does not replace the normal installation guide; it should be used with the normal instructions to make the manual changes needed for a high availability installation.

In an HA configuration, there are two servers:

- one Master Server running the full set of services.
- one Distributed Server running:
  - proxy (sipXproxy)
  - registrar/redirect service (sipregistrar)

Hardware

Both servers should conform to the standard hardware guidelines in the installation guide. However, only the Master Server hosts a VXML service and stores the associated voicemail, so the Distributed Server does not need to budget disk for voicemail; it may have a /var partition that is correspondingly smaller.

Select Server Names and Addresses

Select the fully qualified host names and IP addresses for your sipX servers. In the examples below, “example.com” represents the local domain name; replace it with your actual domain name for your installation. The host names used here are:

- `sipx1.example.com` for the Master Server

- `sipx2.example.com` for the Distributed Server

The HA system as a whole has a name of its own – the SIP domain name. This is the name that is used as the right side of the SIP addresses (sip:user@domain). Typically, this will be the domain name of the organization (example.com), but it may be any unique name within that domain.

Installation

The recommended installation sequence is:

1. Install the Master Server
2. Configure DNS
3. Create TLS/SSL certificates
4. Install the Distributed Server
5. Modify Configurations

Each of these steps is detailed below.

Install Master Server

Install the Master Server software normally, using the names you chose above for the SIP domain name and the host name.

Configure DNS

In order to provide load sharing and failover, all SIP message routing to redundant services in an HA configuration uses DNS SRV records. The following SRV records are required:

domain

The domain record maps the SIP domain name to the proxy host names.

registrar

In an HA configuration, each proxy is configured to use a registry service name that is mapped first to the registrar instance on the same Server, and then falls back to the registrar instance on the other Server .

To generate all the DNS records needed for your installation, use the `sipx-dns` command on the Master Server; log in to the server, and execute:

`sipx-dns` *`sip-domain server-name`*`/server-`*`ip server-name`*`/server-`*`ip`*

substituting your names and IP addresses:

sip-domain
: Is the domain that will be used as the domain part of your SIP addresses. Typically, this will be your top level domain name (example.com).

server-name/server-ip
: This is the fully qualified name and IP address of each sipXpbx server in your domain, separated by a '/' character.

The output of the `sipx-dns` command is suitable for use in a zone file for the BIND nameserver (Linux `named`).

Example DNS Configuration

For the example HA configuration, the DNS records can be generated by:

```
sipx-dns example.com sipx1.example.com/10.1.1.50 sipx2.example.com/10.1.2.50
```

the output generated is:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; sipX Servers for SIP domain 'example.com'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
sipx1.example.com.      IN      A    10.1.1.50
sipx2.example.com.      IN      A    10.1.2.50

example.com.            IN      NAPTR   2 0 "s" "SIP+D2T" "" _sip._tcp.example.com.
example.com.            IN      NAPTR   2 0 "s" "SIP+D2U" "" _sip._udp.example.com.

_sip._tcp.example.com.  IN      SRV  1 0 5060 sipx1.example.com.
_sip._tcp.example.com.  IN      SRV  1 0 5060 sipx2.example.com.
_sip._udp.example.com.  IN      SRV  1 0 5060 sipx1.example.com.
_sip._udp.example.com.  IN      SRV  1 0 5060 sipx2.example.com.

; sipx1.example.com routing for registry/redirect service
_sip._tcp.rr.sipx1.example.com. IN   SRV        1   0 5070 sipx1.example.com.
_sip._udp.rr.sipx1.example.com. IN   SRV        3   0 5070 sipx1.example.com.
_sip._tcp.rr.sipx1.example.com. IN   SRV        2 100 5070 sipx2.example.com.
_sip._udp.rr.sipx1.example.com. IN   SRV        4 100 5070 sipx2.example.com.

; sipx2.example.com routing for registry/redirect service
_sip._tcp.rr.sipx2.example.com. IN   SRV        1   0 5070 sipx2.example.com.
_sip._udp.rr.sipx2.example.com. IN   SRV        3   0 5070 sipx2.example.com.
_sip._tcp.rr.sipx2.example.com. IN   SRV        2 100 5070 sipx1.example.com.
_sip._udp.rr.sipx2.example.com. IN   SRV        4 100 5070 sipx1.example.com.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Create TLS/SSL Certificates

The synchronization of the replicated services in an HA configuration is secured by TLS, and requires that:

- the certificates for each system are signed by an authority trusted by each system.
- Each certificate contains a SubjectAltName entry with the DNS name for the host.

If you use the gen-ssl-keys.sh script to create self-signed certificates for your systems, these requirements are met for you.

Create and install the certificate for the Master Server by logging in to the Master Server, and execute:

```
mkdir $HOME/sslkeys
cd $HOME/sslkeys
/usr/bin/ssl-cert/gen-ssl-keys.sh
```

answer the questions, providing the name of your domain and the Master Server.

To create a compatible certificate for the Distributed Server, in that same directory execute:

```
/usr/bin/ssl-cert/gen-ssl-keys.sh --csr
```

answer the questions; the default answers should be correct for all but the full name of the server : provide the name of the Distributed Server. The above creates a key and a certificate request for the Distributed Server. The certificate request is the file named *dist-server*.csr, where *dist-server* is the name of the Distributed Server. To create the certificate for the Distributed Server, execute:

```
/usr/bin/ssl-cert/gen-ssl-keys.sh   -signdist-server.csr
```

The above will produce the files:

dist-server.key
dist-server.crt
caname.crt

where *dist-server* is the name you provided for the Distributed Server, and *caname* is the name you provided for the Certificate Authority. These three files must be copied to the Distributed Server when it is installed (see below).

Install the certificates and key for the Master Server by running the following as root in that directory:

```
/usr/bin/ssl-cert/install-cert.sh
```

Install the Distributed Server

Install the sipXpbx software on the Distributed Server normally. When you reach the step of generating the TLS/SSL certificate, rather than generating new certificates, copy the files:

```
dist-server.key
dist-server.crt
caname.crt
```

that you created on the Master Server to the Distributed Server and execute the following as root in the directory:

```
/usr/bin/ssl-cert/install-cert.sh
```

When the installation is finished, log in to the servers to modify configurations.

Modify Configurations

There are 5 sets of changes to be made to the configurations:

1. Disable non-replicated services on the Distributed Server.
2. Configure routing for non-replicated services on the Distributed Server
3. Configure the Master Server to manage the Distributed Server.
4. Configure registry synchronization on both servers.
5. Configure routing for redundant services on both servers.
6. Check that the realm is the same on both servers

Disable non-replicated services on the Distributed Server

The Distributed Server runs only three services:

- registrar/redirect service (sipregistrar)
- proxy (sipXproxy)

To disable the other services, changes are needed in the process definition files; these are found in the `/etc/sipxpbx/process.d` directory. There are 3 files in that directory that should be left unmodified:

```
keepalive.process.xml
sipXproxy.process.xml
sipregistrar.process.xml
```

The other files govern processes not needed on a Distributed Server, so the same change is needed in each. These files include (some may not be present in all installations depending on how you installed the original software - a missing file is not a problem):

```
sipstatus.process.xml
sipxacd.process.xml
sipxconfig.process.xml
sipxpark.process.xml
sipxpresence.process.xml
sipXvxml.process.xml
```

Edit each of the above files on the Distributed Server. Each has an entry like this one:

```
<?xml version="1.0"?>
<watchdog-process enable=  true
xmlns="http://www.sipfoundry.org/sipX/schema/xml/watchdogprocess-00-01">
...
</watchdog-process>
```

Modify the attribute to the top level `watchdog-process` element named "`enable`" and set it to "`false`", like this:

```
<watchdog-process enable=  false
xmlns="http://www.sipfoundry.org/sipX/schema/xml/watchdogprocess-00-01">
...
</watchdog-process>
```

Save the file, and then run the following command for each to check that the file is still valid (again, nothing printed means ok):

```
sipx-validate-xml /etc/sipxpbx/process.d/file.process.xml
```

Another way of disabling the services is to just delete each of the above files. This has the disadvantage that the files may re-appear after an update and thus must be deleted again.

Configure routing for non-replicated services on the Distributed Server

In order for requests arriving at the proxy on the Distributed Server to be routed correctly to the services that are available only on the Master Server, edit the file /etc/sipxpbx/config.defs on the Distributed Server.

Change each of the following lines (they are not all together like this in the file):

```
CONFIG_SERVER_ADDR=${MY_IP_ADDR}

MEDIA_SERVER_ADDR=${MY_IP_ADDR}

STATUS_SERVER_ADDR=${MY_IP_ADDR}

VOICEMAIL_SERVER_ADDR=${MY_IP_ADDR}

VOICEMAIL_SERVER_HOSTNAME=${MY_IP_ADDR}

ORBIT_SERVER_ADDR=${MY_IP_ADDR}

PRESENCE_SERVER_ADDR=${MY_IP_ADDR}
```

on each of the above, replace `${MY_IP_ADDR}` with the fully qualified host name of your Master Server.

Configure the Master Server to manage the Distributed Server

On the Master Server, edit the file `/etc/sipxpbx/topology.xml.in`, which should look like:

```
<?xml version="1.0" ?>

<!-- This file defines the "topology" file that resides on the
     configuration server.  This file acts as a registry/database for
     replication targets and remove process management agents -->

<!DOCTYPE topology [
     <!ELEMENT topology (location+)>
     <!ELEMENT location (component+, replication_url, agent_url, sip_domain?)>
     <!ATTLIST location id CDATA #REQUIRED>
     <!ELEMENT component EMPTY>
     <!ATTLIST component id CDATA  #REQUIRED>
     <!ATTLIST component type  (media-server | config-server | comm-server)
#REQUIRED>
     <!ELEMENT replication_url (#PCDATA)>
     <!ELEMENT agent_url (#PCDATA)>
     <!ELEMENT sip_domain (#PCDATA)>
]>

<topology>
   <location id="Config Server, Media Server and Comm Server">
      <component id="MediaServer1" type="media-server" />
      <component id="CommServer1"  type="comm-server" />
      <replication_url>
         https://${MY_FULL_HOSTNAME}:${CONFIG_SERVER_HTTPS_PORT}/cgi-
bin/replication/replication.cgi
      </replication_url>
      <agent_url>
         https://${MY_FULL_HOSTNAME}:${WATCHDOG_SERVER_XMLRPC_PORT}/RPC2
      </agent_url>
   </location>

   <!-- REMOVE THIS LINE TO ADD DISTRIBUTED SERVER
   <location id="Distributed Comm Server">
      <component id="CommServer2"  type="comm-server" />
      <replication_url>
         https://DISTRIBUTED_HOSTNAME:${CONFIG_SERVER_HTTPS_PORT}/cgi-
bin/replication/replication.cgi
      </replication_url>
      <agent_url>
         https://DISTRIBUTED_HOSTNAME:${WATCHDOG_SERVER_XMLRPC_PORT}/RPC2
      </agent_url>
   </location>
   REMOVE THIS LINE TO ADD DISTRIBUTED SERVER -->
</topology>
```

(some lines above are wrapped in this document but not in the original file)

The default file shown instructs the configuration server to replicate configuration information only to itself. To configure replication to the Distributed server, delete the two lines:

```
   <!-- REMOVE THIS LINE TO ADD DISTRIBUTED SERVER
and
   REMOVE THIS LINE TO ADD DISTRIBUTED SERVER -->
```

and change each instance of ‘DISTRIBUTED_HOSTNAME to the full name of your Distributed Server.

Configure registry synchronization on both servers

On both the Master and Distributed Servers, edit the file /etc/sipxpbx/registrar-config.in ; at the bottom of the file, you will find a block of lines:

```
# See HaSetup.pdf for how to use the following
SIP_REGISTRAR_XMLRPC_PORT : 5077
SIP_REGISTRAR_NAME : ${MY_FULL_HOSTNAME}
SIP_REGISTRAR_SYNC_WITH :
```

Change the last line to

SIP_REGISTRAR_SYNC_WITH : *MASTER_HOSTNAME*, *DISTRIBUTED_HOSTNAME*

using the fully qualified host names of your Master and Distributed Servers.

Modify routing for redundant services on both servers

On both the Master and Distributed Servers, edit the file /etc/sipxpbx/config.defs

Find the setting for:

REGISTRAR_SERVER_SIP_SRV_OR_HOSTPORT

and set it to:

REGISTRAR_SERVER_SIP_SRV_OR_HOSTPORT=**rr.${MY_FULL_HOSTNAME}**

Check that the domain and realm are the same on both servers

In `/etc/sipxpbx/config.defs` there are two configuration assignments:

```
SIPXCHANGE_DOMAIN_NAME=`hostname -f`
SIPXCHANGE_REALM=`hostname -d`
```

Each of these assignments must produce the exact same value on both the Master and Distributed servers; that is, the SIPXCHANGE_DOMAIN_NAME must be the same on both servers, and the SIPXCHANGE_REALM must be the same on both servers (the realm may be the same as the domain, but need not be). The eaiest way to ensure this is to remove the hostname commands (including the left-quote characters) and replace them with a string of your choice like:

```
SIPXCHANGE_DOMAIN_NAME=  example.com
SIPXCHANGE_REALM=  Example
```

The domain name is the value used to the right side of the '@' in your SIP addresses (alice@example.com). The realm is used to identify the administrative domain in SIP and web authentication.

Call Resolver High Availability (HA) setup

In an HA environment, call state event logging to a database is done by proxies running on separate machines, with the Call Resolver located on the Master Server. The Call Resolver establishes SSL connections to these databases and resolves the call state events into a single CDR database on the Master Server.

There are a number of configuration steps that must be performed before the Call Resolver can operate on distributed databases. The first step is to follow all of the instructions given in the 'High Availability Setup Guide'. Most important for the subsequent Call Resolver configuration is the creation and installation of SSL certificates on the Master Server and the Distributed Server. The name of the Certificate Authority file (*caname.crt* in the High Availability Setup Guide) is one of the configuration parameters needed by the Call Resolver.

Call Resolver setup on the Master Server

Two configuration parameters in /etc/sipxpbx/callresolver-config.in have to be added to enable the Call Resolver to establish SSL connections to distributed databases :

SIP_CALLRESOLVER_CSE_HOSTS : {comma-delimited list of host names and port numbers}

SIP_CALLRESOLVER_CSE_CA : {name of the Certificate authority file caname.crt}

The Call Resolver uses stunnel to establish SSL connections to remote databases. Stunnel is an SSL-encrypting socket wrapper that provides SSL support to applications by forwarding connections on local ports to another instance of stunnel running on a remote machine.

'localhost' must be specified as part of the list of host names if a CSE database is running on the local machine, otherwise the Call Resolver will not connect to the local CSE database. If no port is specified it will default to the standard PostgreSQL port 5432.

The following example has the Call Resolver running on the Master Server master.example.com and the Distributed Server running on distrib.example.com. The Call Resolver will connect to the database on the Distributed Server on port 5433. The name of the Certificate Authority file is 'ca.example.com.crt'.

SIP_CALLRESOLVER_CSE_HOSTS : localhost, distrib.example.com:5433

SIP_CALLRESOLVER_CSE_CA : ca.example.com.crt

More than one Distributed Server can be specified.

SIP_CALLRESOLVER_CSE_HOSTS : localhost, distrib1.example.com:5433, distrib2.example.com: 5434

SIP_CALLRESOLVER_CSE_CA : ca.example.com.crt

Note that the ports specified can be any unused ports on the Master Server. The local PostgreSQL database is using its default port 5432, so 5433 and 5434 are unused but 'PostgreSQL-like' ports). The specified ports do not describe any real connections to distributed systems, they are only used for forwarding local connections on these ports to the distributed machines. The port numbers must be unique, no two host names can have the same port number.

Distributed Server setup

Because the web UI can only configure proxies on the Master Server the configuration files on the Distributed server need to be edited manually. To enable call state event logging manually edit the file /etc/sipxpbx/proxy-config.in to include the line

SIP_PROXY_CALL_STATE_DB : ENABLE

and the file /etc/sipxpbx/authproxy-config.in to include the line

SIP_AUTHPROXY_CALL_STATE_DB : ENABLE

in order to enable call state logging to the database.

Since the Call Resolver is not running on the Distributed Server there is no need to configure it. The Distributed Server still must be configured to accept SSL connections from the Call Resolver running on the Master Server to allow the Call Resolver to retrieve call state events.

The following example uses the Certificate Authority file 'ca.example.com.crt' to set up the Distributed Server. As root from the command line enter the command

sipxha-distrib.sh setup ca.example.com.crt

and the Distributed Server should be set up to accept SSL connections to its database.

Setup Complete

Your HA setup is now complete.

When your system starts, you should see lines like these near the beginning of the /var/log/sipxpbx/sipregistrar.log file (it's easier to read if you use the syslogviewer command):

```
# syslogviewer -i < /var/log/sipxpbx/sipregistrar.log
2006-02-23T22:31:53.540094Z"
   :3:SIP:NOTICE:sipx1.example.com::00000000:SipRegistrar:
   SipRegistrar >>>>>>>>>>>>>>> STARTED
2006-02-23T22:31:53.542391Z"
   :19:SIP:NOTICE:sipx1.example.com::00000000:SipRegistrar:
   SipRegistrar::configurePeers: sipx2.example.com
2006-02-2006-02-23T22:31:53.777835Z"
   :95:SIP:NOTICE:sipx1.example.com::B6BF5BB0:SipRegistrar:
   registerSync.reset success to 'sipx2.example.com'; update numbering
synchronized.
```

There is an extra line wrap in the above after the timestamp to make the messages fit better on the page, and there may be more messages in your output depending on your configuration. The 'configurePeers' line shows what hosts are configured as peers, and registerSync.reset line shows that the initial synchronization with that peer has been achieved. Since the two systems cannot come up exactly simultaneously, the first to come up will show an error when it first

attempts to contact its peer, indicating that the peer is UnReachable. This is normal, and will be corrected as soon as the peer initializes; there will be a log entry showing when that occurs to indicate that update numbering is synchronized.