SOUNZ Standby HOWTO

Author: Paul Waite
Created: 28th October 2007

Contents

Glossary of Terms

Primary Server - a physical machine is deemed to be the primary SOUNZ server. It remains 'primary' even if it is running in Standby mode, or is dead. Obviously it is the machine which is intended to be the usual Production Server.

Secondary Server - As above, except the physical machine which is intended to be the Standby Server.

Production Server - The server which is currently serving applications for SOUNZ users.

Standby Server - The server which can be switched to Production if the current Production server has a problem.

1. Overview

SOUNZ has two machines, at least one of which is the Production server at any one time. The other machine is the Standby Server, and is normally used as a Staging Server to test enhancements or other changes.

NOTE: the file **`/etc/sounz/sounz.conf`** contains two important settings:

```
IP_PRIMARY_SERVER="slayer.servers.catalyst.net.nz"
IP_SECONDARY_SERVER="megadeth.servers.catalyst.net.nz"
```

These MUST be set to the relevant *internal network* IP address/hostname of the designated primary (nominal Production) and secondary (nominal Standby) machines. All of the various standby scripts use these values, so please make sure that you check these if the IP addresses/hostnames change.

By internal network we mean the Catalyst management interface IP's (or hostnames) should be used, since these will never change. In SOUNZ case these will be the hostnames shown above.

To keep the Standby server up to date, and hence ready to take over the Production if we have a hardware failure the following stuff happens, driven from cron, in the small AM hours:

1. Postgres database is dumped regularly on the Production server.
2. The Apollo index is replicated to the Standby server using rsync.
3. Standby Server rsyncs Production application data areas (/data/sounz).
4. Standby Server rsyncs Production application home areas (/usr/share/sounz).
5. The Standby server replaces it's Postgres database with the latest snapshot.

Back to Contents

---

2. SOUNZ Server Statuses

Both servers are deemed to be in a 'status'. This status is determined by the content of the status file **`/etc/sounz/sounz.status`** and it may contain one of the below values. These statuses are used in various scripts to make sure they do the right thing.

down
Means the server is currently not working.

production
Normal mode
The server is the Production server, serving user applications, and is running alongside a working Standby server.

standby
Normal mode
The server is the Standby server, and is running alongside a working Production server.

production_alone
Emergency mode
The server is the Production server, serving user applications, and is running alone. This is usually when the former Production server has croaked, and this server has been switched from '**standby**' status to '**production_alone**' in an emergency situation.

standby_as_production
Recovery mode
The server is the former Standby server, running as Production, serving user applications, and is running alongside a former Production server which is now a Standby server. In other words - both servers are running in opposite roles to normal.

production_as_standby
Recovery mode
The server is the former Production server, running as a Standby server, and is running alongside a former Standby server which is running as the Production server.

Back to Contents

---

3. PROCEDURES

3.1 EMERGENCY - Switching Standby Server To Production

If the production server croaks, then you will want to get the Standby server up and running pretty damn quick. In that case, perform the following steps:

Step 1: Switch Production server into Standby mode

If the production server is accessible, then log in and do the following, as root:

- cd /usr/share/sounz/scripts/dr
- ./switch-to-standby.sh
- From the menu, choose '1' the 'EMERGENCY' option.
- After that, the script will set the status to '**down**'.

Step 2: Fix up Ethernet Interfaces and IP's

Bring down *eth0* on the Production server which will normally be configured as 202.78.240.46

Go across to the Standby server and bring that IP up as *eth0:1*

Example: if **slayer** is the Production Server on *eth0:202.78.240.46*, and **megadeth** is the Standby Server on *eth0:202.78.240.47*, then *eth0* on **slayer** would be brought down, and on **megadeth** an *eth0:1* would be brought up as *202.78.240.46*. In that way, **megadeth** will start serving the application website in the place of **slayer**, with no DNS changes required.

Step 3: Switch Standby server into Production mode

As the root user, do the following on the Standby server:

- cd /usr/share/sounz/scripts/dr
- ./switch-to-production.sh
- From the menu, choose '1' the 'EMERGENCY' option.

- It then asks if you want to grab the latest data from Production. Obviously do so if you can, but this depends on the state of Production of course.

- It should confirm that you are in '**production_alone**' status.

Step 5: Run some checks

Browse to http://sounz.org.nz and check that the website front page is served.

Apollo should be running as a java daemon and listening on port 8983. You can find the process by doing this ps command: **ps ax|grep start.jar**

Examine the SOUNZ config in **/etc/sounz/sounz.conf** and find the `MONGREL_PORTS` variable setting. There should be mongrel servers listening on all of those ports. Find these processes by the following ps command: **ps ax| grep ruby|grep script/server**

Postgresql should be running locally and there should be a database called **sounz** defined and available to local connections, both host (tcp) and direct.

There should be two rsync daemons running. One is for apollo (Solr) and it should be on the port defined in **/etc/apollo/apollo.conf** (usually port 18983). The other is for the SOUNZ application itself, and it will be running on the port defined as **SOUNZ_RSYNC_PORT** in **/etc/sounz/sounz.conf** (usually port 18993). You can find these processes with this ps command: **ps ax|grep rsync**

Tail the following logs for activity:

- /var/log/sounz/sounz-app/production.log (application activity on each website hit)
- /var/log/apollo/apollo.log (Solr indexing/querying activity)
- /var/log/sounz/apache2/sounz.org.nz-access.log
- /var/log/sounz/apache2/sounz.org.nz-error.log
- /var/log/sounz/sounz-general.log (mainly replication cronjob and rsync activity)

Back to Contents

3.2 RECOVERY - Bringing Primary Machine Back As Production

Having done all the emergency stuff above, you probably went and fixed the hardware on the duff production server, and you now want to get it back into the groove and let the Standby server go back to quaffing beers and watching TV with its feet up on the couch again. First, let's revisit our glossary of terms for this. Remember:

The *secondary server* is the machine which is the Standby server currently running as production.

The *primary server* is the machine which had the problem, and has just been mended.

Ok, here's what you do..

Step 1: Switch primary server into Standby Recovery mode

As root on the primary server (the one you just fixed up)

- cd /usr/share/sounz/scripts/dr
- ./switch-to-standby.sh
- From the menu, choose '3' the Recovery option.

In actuality, nothing much should change, except the machine status gets set to 'standby'.

Step 2: Switch secondary server into Production Recovery mode

As root on the 'secondary server' (the one running as Production)

- cd /usr/share/sounz/scripts/dr
- ./switch-to-production.sh
- From the menu, choose '3' the Recovery option.

Once again, only the status is changed, so that replication scripts do the right thing.

At this point we have the following situation - the primary server, which you fixed, is now acting as a Standby server. The secondary server is still handling clients. In short, the system is as it was before, but with the roles of each machine reversed.

Monitor this intermediate situation for however long you need, to determine that the primary machine is permanently fixed and is going to be ok to use.

Step 3: Switch secondary server into Standby mode

As root on the 'secondary server' (the one running as Production)

- cd /usr/share/sounz/scripts/dr
- ./switch-to-standby.sh
- From the menu, choose '2' the Normal option.

When this is done, your secondary machine will be in '**standby**' status.

Step 4: Fix up Ethernet Interfaces and IP's

Do whatever 'magic' is required with LAN interfaces to make the primary machine once again the one users will be accessing. Normally this is simply bringing the temporary *eth0:1* interface down on the secondary box, and bringing the *eth0* interface back up on the primary box.

Step 5: Switch primary server into Production mode

As root on the 'primary server' (the one you fixed)

- cd /usr/share/sounz/scripts/dr
- ./switch-to-production.sh
- From the menu, choose '2' the Normal option.

This will set this machine into '**production**' status, and once again replication will be going from the primary machine --> secondary.

Back to Contents
