

Enterprise Deployment with Corosync

About

by [Emmanuel Schmidbauer2](#)

This uses the same concept discussed [here](#), which I also wrote.

This setup was tested on Debian Squeeze. There are two example servers here, fs01-a (10.10.10.8) and fs01-b (10.10.10.9).

10.10.10.11 will be the floating IP used here. This tutorial assumes you have ODBC setup so both nodes can connect to the same database. Please note that 10.10.10.11 should be a public facing IP, it is only used as an example here.

Network Configuration

Add each node to hosts file

```
root@fs01-a:~# echo "10.10.10.8 fs01-a" >> /etc/hosts
root@fs01-a:~# echo "10.10.10.9 fs01-b" >> /etc/hosts
root@fs01-b:~# echo "10.10.10.8 fs01-a" >> /etc/hosts
root@fs01-b:~# echo "10.10.10.9 fs01-b" >> /etc/hosts
```

Create SSH keys

```
root@fs01-a:~# ssh-keygen -t dsa -f ~/.ssh/id_dsa -N ""
root@fs01-b:~# ssh-keygen -t dsa -f ~/.ssh/id_dsa -N ""
```

Create SSH key to other server

```
root@fs01-a:~# scp -P 22 .ssh/id_dsa.pub root@fs01-b:/root/
root@fs01-a:~# cat id_dsa.pub >> .ssh/authorized_keys
root@fs01-a:~# rm -rf fs01-b.pub
root@fs01-b:~# scp -P 22 .ssh/id_dsa.pub root@fs01-a:/root/
root@fs01-b:~# cat id_dsa.pub >> .ssh/authorized_keys
root@fs01-b:~# rm -rf fs01-a.pub
```

Allow FreeSWITCH to bind to nonlocal IP

Add the following line to /etc/sysctl.conf

```
echo 'net.ipv4.ip_nonlocal_bind=1' >> /etc/sysctl.conf
```

Restart networking:

```
/etc/init.d/networking restart
```

Run:

```
sysctl -p
```

You should see:

```
net.ipv4.ip_nonlocal_bind = 1
```

If you don't, you did something wrong, if you do, continue...

Corosync/Pacemaker

Install Corosync & Pacemaker

```
apt-get install pacemaker crmsh -y
```

corosync is included as a dependency of pacemaker

Configure Corosync

```
root@fs01-a:~# export ais_port=4000
root@fs01-a:~# export ais_mcast=226.94.1.1
root@fs01-a:~# export ais_addr=`ip addr | grep "inet " | tail -n 1 | awk '{print $4}' | sed s/255/0/`
root@fs01-a:~# cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf
root@fs01-a:~# sed -i.bak "s/.*mcastaddr:.*mcastaddr:\ $ais_mcast/g" /etc/corosync/corosync.conf
root@fs01-a:~# sed -i.bak "s/.*mcastport:.*mcastport:\ $ais_port/g" /etc/corosync/corosync.conf
root@fs01-a:~# sed -i.bak "s/.*\tbindnetaddr:.*bindnetaddr:\ $ais_addr/g" /etc/corosync/corosync.conf
root@fs01-a:~# cat >> /etc/corosync/corosync.conf <<EOT
    aisexec {
        user: root
        group: root
    }
    service {
        name: pacemaker
        ver: 0
    }
EOT
root@fs01-a:~# corosync-keygen
root@fs01-a:~# scp -P 22 /etc/corosync/authkey root@fs01-b:/etc/corosync/authkey
root@fs01-a:~# scp -P 22 /etc/corosync/corosync.conf root@fs01-b:/etc/corosync/corosync.conf
root@fs01-a:~# sed -i "s/no/yes/g" /etc/default/corosync
root@fs01-b:~# sed -i "s/no/yes/g" /etc/default/corosync
```



OpenAIS is deprecated in favor of corosync.

<https://github.com/corosync/openais>

The LSB (Linux Standards Base) Script

Copy and paste this script into /etc/init.d/FSSofia on both nodes.

```

#!/bin/sh
### -*- mode:shell-script; indent-tabs-mode:nil; sh-basic-offset:2 -*-
### BEGIN INIT INFO
# Provides: FSSofia
# Required-Start: $network $remote_fs $local_fs
# Required-Stop: $network $remote_fs $local_fs
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: FSSofia
# Description: FSSofia Status
### END INIT INFO
#set -x

FS_CLI_PROG='/usr/local/freeswitch/bin/fs_cli'
FS_CLI_HOST='127.0.0.1'
FS_CLI_PORT='8021'
FS_CLI_PASS='ClueCon'
PROFILES='internal'

usage() {
    echo "Usage: $0 profile1[,profile2[,etc]] {start|stop|status}"
    exit 1
}

fs_cli() {
    $FS_CLI_PROG -H $FS_CLI_HOST -P $FS_CLI_PORT -p $FS_CLI_PASS -x "$1"
}

sofia_profile_started() {
    fs_cli "sofia xmlstatus" | grep "<name>$1</name>" | wc -l
}

if [ $# != 1 ]; then
    usage
fi

#PROFILES=`echo $1 | tr ',' ' '`
CMD=$1
#was $2

case "$CMD" in
    'start')
        fs_cli "sofia recover"
        exit 0
        ;;
    'stop')
        exit 0
        ;;
    'status')
        for p in $PROFILES; do
            if [ `sofia_profile_started "$p"` -eq 0 ]; then
                echo "$p DOWN"
                exit 3
            fi
        done
        echo "OK"
        exit 0
        ;;
    *)
        usage
        ;;
esac

```

Make the script executable

```
root@fs01-a:~# chmod +x /etc/init.d/FSSofia
root@fs01-b:~# chmod +x /etc/init.d/FSSofia
```

Restart Corosync

```
root@fs01-a:~# /etc/init.d/corosync restart
root@fs01-b:~# /etc/init.d/corosync restart
```

crm configure show

Configure corosync/pacemaker as follows

```
root@fs01-a:~# crm configure edit
```

```
node fs01-a \
    attributes standby="off"
node fs01-b \
    attributes standby="off"
primitive fs lsb:FSSofia \
    op monitor interval="1s" enabled="true" timeout="2s" on-fail="standby" \
    meta target-role="Started"
primitive fs-ip ocf:heartbeat:IPaddr2 \
    params ip="10.10.10.11" nic="eth0:0" cidr_netmask="24" \
    op monitor interval="10s"
group cluster_services fs-ip fs
location cli-prefer-cluster_services cluster_services \
    rule $id="cli-prefer-rule-cluster_services" inf: #uname eq fs01-a
property $id="cib-bootstrap-options" \
    dc-version="1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    stonith-enabled="false" \
    last-lrm-refresh="1348755080" \
    no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

Configure FreeSWITCH

Set FreeSWITCH to listen on the floating IP and configure ODBC.

sofia.conf.xml

You should have the following parameters set in sofia.conf.xml on both nodes. The Database needs to be shared between the two+ nodes. Please see the databases wiki for the proper odbc connection string.

```
<param name="odbc-dsn" value="database:username:password"/>
<param name="track-calls" value="true"/>
<param name="rtp-ip" value="10.10.10.11"/>
<param name="sip-ip" value="10.10.10.11"/>
<param name="presence-hosts" value="10.10.10.11"/>
<param name="ext-rtp-ip" value="10.10.10.11"/>
<param name="ext-sip-ip" value="10.10.10.11"/>
```

Configure sip_profiles

You should edit all sip profiles that you wish to use (typically internal.xml and external.xml). Adding the following line to tell these modules how to connect to your shared database and to track the active calls.

```
<param name="odbc-dsn" value="database:username:password"/>
<param name="track-calls" value="true"/>
```

switch.conf.xml

You should have the following parameters set in switch.conf.xml on both nodes

```
<param name="switchname" value="fs01"/>
<param name="core-db-dsn" value="database:username:password"/>
<param name="core-recovery-db-dsn" value="database:username:password"/>
```

Other XML Files

These are not necessary for everyone that wants HA. But you may need to add in the "odbc-dsn" parameter depending on what you are using freeswitch for:

```
# Add the following line in db.conf.xml, voicemail.conf.xml, lcr.conf.xml, and nibblebill.conf.xml
<param name="odbc-dsn" value="database:username:password"/>
```

Start FreeSWITCH

Start FreeSWITCH on both nodes.

```
root@fs01-a:~# /etc/init.d/freeswitch start
root@fs01-b:~# /etc/init.d/freeswitch start
```

Start Corosync

Start Corosync on both nodes.

```
root@fs01-a:~# /etc/init.d/corosync start
root@fs01-b:~# /etc/init.d/corosync start
```

Start Pacemaker

Start Corosync on both nodes.

```
root@fs01-a:~# /etc/init.d/pacemaker start
root@fs01-b:~# /etc/init.d/pacemaker start
```

TEST!

Make a call and don't hang up.

CLI on both FS nodes

```
root@fs01-a:~# /usr/local/freeswitch/bin/fs_cli
root@fs01-b:~# /usr/local/freeswitch/bin/fs_cli
```

Crash the primary server

```
freeswitch@default> fsctl crash
```

Voilà, the call should continue almost instantly on the standby server!

To Do

TODO: Fill in how to setup/install corosync

See also

[Enterprise Deployment](#)

[High Availability](#)

[Clusters From Scratch](#) (Pacemaker, Corosync home; Red Hat)

[Debian Clusters](#)

[crmsh](#) - the cluster management shell (alternative to pcs)

[SmartVox Discussion of Pacemaker with OpenSIPS](#)