

HA keepalived

About

Example configuration for setting up Freeswitch using Keepalived to manage the IP address failover.

This was done due to heavy complexity of the corosync/pacemaker approach, and difficulties getting that setup to actually fail over properly in response to a 'fsctl crash'.

Follow the instructions in the [Freeswitch High Availability](#) page to set up call tracking, odbc, etc.

Topology

This example has both a public network (on eth0) and a private/internal network (on eth1).

```
Node 1 Public: 10.0.0.1
Node 2 Public: 10.0.0.2
Public Virtual IP: 10.0.0.10
Initially Master
```

```
Node 1 Internal: 10.1.0.1
Node 2 Internal: 10.1.0.2
Internal Virtual IP: 10.1.0.10
Initially Standby/Backup
```



In this example FreeSWITCH is installed to /local/freeswitch/server, so adjust paths accordingly.

Perl Scripts

/local/freeswitch/ka-status.pl

ka-status.pl

```
#!/usr/bin/perl

use Sys::Syslog;
openlog "ka-status", "ndelay,pid", "local0";

my @required = ("internal");

my %saw = ();
open(my $in, "-|") || exec("/local/freeswitch/server/bin/fs_cli", "-x", "sofia xmlstatus");
while ( defined(my $line = <$in> ) )
{
    if ( $line =~ m|<name>(.*?)</name>|o )
    {
        $saw{$1} = 1;
    }
}
close($in);

foreach my $profile ( @required )
{
    if ( ! $saw{$profile} )
    {
        syslog(LOG_INFO, "sip profile $profile not found, marking failure");
        exit(1);
    }
}

syslog(LOG_INFO, "all required sip profiles found, marking success");
exit(0);
```

/local/freeswitch/ka-notify.pl

ka-notify.pl

```
#!/usr/bin/perl

# INSTANCE|VI_FREESW|BACKUP|50
my ($what,$id,$state,$prio) = @ARGV;
open(STDOUT, "|/usr/bin/logger -t ka-notify");

print "what($what) id($id) state($state) prio($prio)\n";

if ( $state eq "MASTER" )
{
    print "Instance went to master, issuing sofia recover.\n";
    system("/local/freeswitch/server/bin/fs_cli", "-x", "sofia recover");

    # Not ideal, but it seems to fail over too quickly for skinny devices
    # and since they don't actually handle the failover, need to poke them
    # Comment this out if you're not using mod_skinny/SCCP
    system("/local/freeswitch/server/bin/fs_cli", "-x", "reload mod_skinny");
}
```

Node Configurations

Node 1

/etc/keepalived/keepalived.conf

keepalived.conf

```
global_defs {
    notification_email {
        your_email_address
    }
    smtp_server smtp_server_ip_or_hostname
    smtp_connect_timeout 30
    router_id FREESW
}

vrrp_script chk_fs {
    script "/local/freeswitch/ka-status.pl"
    interval 10
}

vrrp_script chk_fs_port {
    script "</dev/tcp/127.0.0.1/8080"
    interval 1
}

vrrp_instance VI_FREESW {
    state MASTER
    interface eth0
    virtual_router_id 51
    # higher is preferred for master
    # disable to have failover be sticky
    #priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass xxxxxxxxxxxxxxxxxxxx
    }
    notify "/local/freeswitch/ka-notify.pl"
    virtual_ipaddress {
        10.0.0.10 dev eth0
        10.1.0.10 dev eth1
    }
    track_script {
        chk_fs
        chk_fs_port
    }
    smtp_alert
}
```

Node 2

/etc/keepalived/keepalived.conf

keepalived.conf

```
global_defs {
    notification_email {
        your_email_address
    }
    smtp_server smtp_server_ip_or_hostname
    smtp_connect_timeout 30
    router_id FREESW
}

vrrp_script chk_fs {
    script "/local/freeswitch/ka-status.pl"
    interval 10
}

vrrp_script chk_fs_port {
    script "</dev/tcp/127.0.0.1/8080"
    interval 1
}

vrrp_instance VI_FREESW {
    state BACKUP
    interface eth0
    virtual_router_id 51
    # higher is preferred for master
    # disable to have failover be sticky
    #priority 50
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass xxxxxxxxxxxxxxxxxx
    }
    notify "/local/freeswitch/ka-notify.pl"
    virtual_ipaddress {
        10.0.0.10 dev eth0
        10.1.0.10 dev eth1
    }
    track_script {
        chk_fs
        chk_fs_port
    }
    smtp_alert
}
```