High Availability on MySQL service faillure

Contents

Overview

This document outlines the steps required to configure Swivel clusters to trigger fail-over on MySQL failure.

Originally any Swivel HA cluster is configured to fail-over on Tomcat service failure; this will add MySQL service motorization to trigger fail over.

Configuration Considerations

MySQL is extremly robust. If MySQL fails, the probability of having an underlying bigger problem (that could lead to database corruption) is high.

For this reason, the configuration doesn't perfectly mimmic the Tomcat Failure setting. In truth, the Tomcat setting is set not only to fail-over to a different node when one fails, but also to look for the primary node to come up, and then shift load to it.

Due to the causes of a MySQL service faillure, this "shift back to primary node" behaviour is not set.

Swivel configuration for both nodes

1- Enter the Swivel console, and then follow to command line, using Advanced menu (option 8), and then Command line (option 6).

2- To install the correct software packages that will support the monitoring functions, run command:

rpm -ivh http://vault.centos.org/4.9/centosplus/i386/RPMS/mysqlclient14-4.1.22-1.el4s1.1.i386.rpm

3- We need to update the mysql.monitor file by running the command:

wget ?qN http://yum.swivelsecure.net/upgrades/mysql.monitor -O /usr/lib/mon/mon.d/mysql.monitor

4- Add executable permitions to the newlly updated file, by running command:

chmod +x /usr/lib/mon/mon.d/mysql.monitor

This next step can be performed directly by editing the files, or using webmin to do that form you.

Option A - Command line

1- Enter the Swivel console, and then follow to command line, using Advanced menu (option 8), and then Command line (option 6).

2- Run command:

nano /etc/mon.cf

3- Add the LocalHost hostgroup, and then add the LocalHost_IP watch section exemplified bellow.

```
[admin@primary mon]# more mon.cf
# Swivel Appliance Build primary mon.cf file
### global options
cfbasedir    = /etc/mon
pidfile      = /var/run/mon.pid
statedir     = /var/lib/mon/state.d
logdir       = /var/lib/mon/log.d
dtlogfile    = /var/lib/mon/log.d/downtime.log
alertdir     = /usr/lib/mon/alert.d
mondir       = /usr/lib/mon/mon.d
maxprocs     = 20
histlength   = 100
randstart    = 30s
authtype     = pam
userfile     = /etc/mon/userfile

### group definitions (hostnames or IP addresses)
hostgroup Primary_IP 192.168.114.36
hostgroup Standby_IP 192.168.11.37
hostgroup Virtual_IP 192.168.114.38
hostgroup DR_IP 192.168.0.35
hostgroup LocalHost_IP 127.0.0.1

#
# PINsafe
#
watch Primary_IP
    service tomcat5
        description PINsafe Monitoring
        interval 10s
        monitor https.monitor -p 8080 -u /pinsafe/AgentXML?xml=%3CSASF
E
        period
            alert ha.alert
            alertafter 3 120s
            numalerts 1

watch LocalHost_IP
    service mysql
        interval 10s
        monitor mysql.monitor --database=pinsafe_rep
        period
            alert ha.alert

watch DR_IP
    service tomcat5
        description PINsafe Monitoring
        interval 5m
        monitor https.monitor -p 8080 -u /pinsafe/AgentXML?xml=%3CSASF
E
        period
            alertafter 3 20m
            alertevery 5m
            numalerts 10
            alert mail.alert -S "DR is Down - PINsafe" root@localhost
```

4- Use ctrl+o to write the changes to the file, and ctrl-x to quit nano.

5- Use the 'exit' command to quit the command line and return to the appliance menu.

6- use 0 to return to the main menu and then use command 3 to enter Monitor service control

7- use 1 to stop monitor and then 1 to start it back on.

Option B - Webmin

1- Login to Webmin on https://serverIP:10000

2- Goto SYSTEM menu, then MON menu

3- Select Host groups icon, and then add the LocalHost_IP group to the last line, just as shown in the picture

Host Groups

Group name	Member hosts
Primary_IP	192.168.114.36
Standby_IP	192.168.11.37
Virtual_IP	192.168.114.38
DR_IP	192.168.0.35
LocalHost_IP	127.0.0.1

Save

Return to MON index

4- On the watchlists menu, add a new watch list for Localhost, then add a service watch and configure it to:

- Name of service: musql

- Check every: 10 seconds

- Standard Monitor: mysql.monitor

- Monitor parameters: --database=pinsafe_rep

- Alerts for period: ha.alert on Service goes down

...according to the following image.

Edit Service

Watched service details

Name of service	mysql
Description	
Using monitor	⊙ Standard monitor [mysql.monitor ▾] ○ Other monitor ..
Monitor parameters	--database=pinsafe_rep

Check ever

Monitoring period ?

⊙ Specified days and hours   Days to check ⊙ All ○ [Mon ▾] - [Mon ▾]   Hours to check ⊙ All ○ [   ] - [   ]

○ Time::Period string   [   ]

Alerts for period

Alert	Run when	Additional parameters
ha.alert ▾	Service goes down ▾	
▾	Service goes down ▾	

Send alert	⊙ Every time monitor is run ○ Every [ ] [seconds ▾]
Failures before alert	⊙ Immediately ○ After [ ] failures Within time interval [ ] [seconds ▾]
Maximum alerts to send	⊙ Unlimited ○ [ ]

[Save]

◀ Return to watch lists

5- return to MON services and choose MON service restart

6- The final result on MON Status, should be something like this:

MON Status

MON: Operation Status: Summary View

Show Operational Status (summary)   Show Alert History   Load scheduler state   Start scheduler
Show Operational Status (full)   Show Downtime Log   Save scheduler state   Stop scheduler

This information was presented at 13:01:38 on Friday, 25-Apr-2014
The scheduler on localhost:2583 is currently . This page will reload every

Host Group	Service (legend)	
DR_IP	RADIUS	-2m47s (Last OK:
LocalHost_IP		-1s
Primary_IP		+0s

Service color legend:
(top of table)

Unchecked   Good   Failed
(no alerts sent)

Show Operational Status (summary)   Show Alert History   Load scheduler state   Start scheduler
Show Operational Status (full)   Show Downtime Log   Save scheduler state   Stop scheduler

For questions about this server,
contact BOFH@your.domain

◄ Return to MON index

Swivel configuration to support the new HA settings

So far, we managed to setup the HA routines on Swivel secure appliance cluster to faill-over when the MySQL service stops responding.

By default it is setup to do so when the Tomcat service fails, bringing the RADIUS server down with it. However, in the event of a MySQL service failure without Tomcat failure, Radius will still be functional and responding to requests. This will generate an error, because most swivel integrations support radius servers as primary and secondary RADIUS. If the RADIUS service on the down machine accepts the RADIUS request, the integrated client hardware or software will not try the secondary RADIUS. The primary RADIUS however will fire an database connectivity error due to fact that the local host mysql service is down.

To avoid that, we need to assure that both appliance nodes connect to running MySQL service, and that is the on indicated by the appliance Virtual IP.

On both nodes:

1- Enter the Swivel web console, and then navigate to Database.

2- Choose General from the menu, and then edt the MySQL5 connection, replacing Localhost by the VIP assigned to the cluster, on the MYSQL5 connector URL setting

![SWIVEL the power of knowing]

- Status
- Log Viewer
- Server
- Policy
- Logging
- Transport
- Database
  - General
  - MySQL 5
  - Connection Pool
- Mode
- Repository
- RADIUS
- Migration
- Appliance
- OATH
- Synchronisation Administration
- Reporting
- User Administration
- Save Configuration
- Administration Guide
- Logout

Database>General ⑩

Please select and configure a Database. The selected Database will be used to hold authentication i

Database: MySQL 5   ▼

Case sensitive usernames: No ▼

Databases:

⊞ Shipping

⊞ Internal

⊞ JDBC

⊞ MS SQL Server

⊟

     Identifier: MySQL 5

     Class: com.swiveltechnologies.pinsafe.server.user.databa

     Driver: com.mysql.jdbc.Driver

     URL: jdbc:mysql://localhost/pinsafe_rep

     Username: pinsafe

     Password: »»»»»»»»»»»»»»»»»»»»»

⊞ Oracle 10g

⊞ Appliance Database

⊞ New Entry