GUMS High Availability Installation & Configuration Guides

Dan Yocum, Steven Timm
Fermi National Accelerator Laboratory

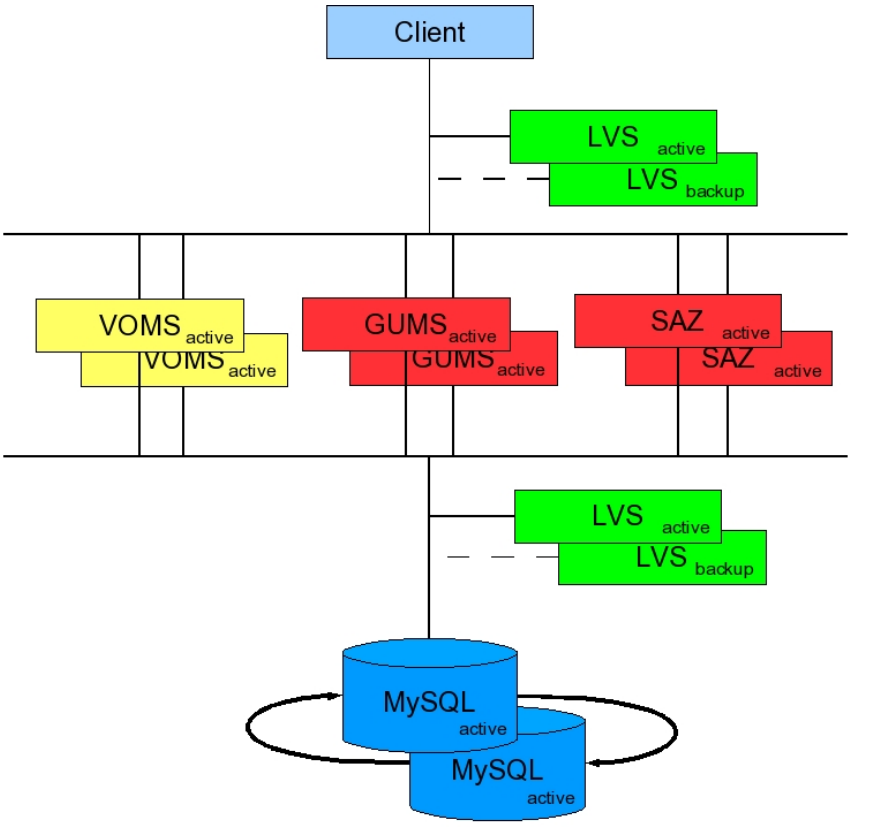
Last Edited on August 5, 2009

Introduction

This document is a modified version of the FermiGrid High Availability Installation and Configuration Guide. It is meant only to configure GUMS and MySQL. It is prepared for
the OSG Site Admins Tutorial. All original writing was done by D. Yocum, adaptation for GUMS only was done by S. Timm. Original intro follows below.

This set of guides is meant to be a concise set of recipes to install and configure the FermiGrid High Availability authentication and authorization middleware system. They are not meant to be detailed guides that cover every possible alternative method of installation and configuration. The goal of these guides is produce a Highly Available, Fault Tolerant system in the shortest amount of time with the least amount of effort. If more detail is required than what is provided herein, that exercise is left to the user. For your convenience, links to those external guides are provided.

The latest version of this document can always be found here:
http://docs.google.com/Doc?docid=0AVY11NMVz7mPZGdmNjg0bWZfMmZiNWttbmQ0&hl=en



Note 1: All network connections are on the public network
Note 2: LVS directors displayed separately for convenience – they are the same in reality

---

Acknowledgements

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359.

Table of Contents

- Xen Installation & Configuration
  - Scientific Linux v5 Installation
  - Xen Installation
  - Xen Configuration

  Circularly Replicated MySQL Installation & Configuration
  - Master Installation
    - Recovery from Server or Replication Error
  - Slave Installation
  - Restart Procedure after Unclean Shutdown
  - Logrotate mysqld.conf
  - Configure iptables
  - MySQL Upgrade Procedure
  - Troubleshooting an Upgrade
- Piranha Linux Virtual Server Installation & Configuration

    - LVS Installation
    - Configure iptables
    - Configure Virtual Servers
- GUMS Installation & Configuration
    - GUMS Installation
    - Configure iptables
- Troubleshooting
- Appendix
    - iptables for MySQL
    - iptables for LVS
    - iptables for VOMS
    - iptables for GUMS
    - iptables for SAZ
    - LVS lvs.cf file
    - LVS Check MySQL script
    - LVS Check VOMS Admin script
    - LVS Check GUMS script

Xen Installation and Configuration

This section describes the procedures to install and configure Xen v3.2.0 on a Scientific Linux v5 system.

Scientific Linux Fermi 5 Installation and Configuration

Download a CD boot.iso image from ftp://linux.fnal.gov/linux/slf53/x86_64/sites/Fermi/images/boot.iso.
Burn the CD using the command cdrecord -v driveropts=burnfree -tao -speed=4 -eject -dev=ATA:1,0,0 -pad -padsize 30s boot.iso
Insert the CD into the CD Rom drive of the computer and boot it.

Or by using dd, write the boot.iso to a USB stick.
At the prompt, enter "http text" This will automatically make sure you install Scientific Linux Fermi
(as opposed to Scientific Linux) from the right server.

Network: Enable IPV4, manual IP configuration. Disable IPV6

Partitioning: The / partition should be 10GB, the swap be at least 8 GB, and one extra partition of 10GB should be
made in the base instance for /linux32. If space is available, a 10GB /usr/local and/or a 10GB /var partition can be made.
All other disk should be made into an LVM partition (type 8E) and you can assign the sub-

partitions once you get the
system up.

Workgroup: Choose "FermiGrid workgroup install"

Grub Password: You should include a grub password, make it the same as the root password.

For the HA systems, we do not select the virtualization package, instead we bring our own along after the base install is done.
The install will proceed apace. After it is done, it will reboot and it will come up to a text firstboot menu.
On this menu, you need to go into firewall configurations, disable the firewall, and turn off SELinux. You should
also go through the authentication menu although it is not usually necessary to change anything.

Once firstboot is done, you get to a login prompt. Log in to the system and do the following:

1) copy the stock passwd, group, auto.master, auto.home, auto.grid, and auto.ilc files from fermigrid1.
2) copy the /root/.k5login file from fermigrid1
3) copy the kerberos keytab from the stock tree on fermigrid0 where all keytabs are stored, into /etc/krb5.keytab.
4) Be sure sshd is running, verify that you can log into the machine.
5) get SLF4 krb5-workstation-fermi and krb5-libs-fermi if you need to run kcron

Xen Installation (Xen 3.1.2/SL5.3)

1) yum install xen xen-devel kernel-xen virt-manager
2) Modify /etc/sysconfig/kernel, change the default kernel to kernel-xen. This makes sure that kernel-xen will be set as the default kernel in grub.conf in future.
3) Check that grub.conf is booting the kernel-xen with the right console parameters.

```
[root@fermigrid2 ~]# more /boot/grub/grub.conf
serial --unit=0 --speed=115200
terminal --timeout=10 serial console
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You do not have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /, eg.
```

```
#         root (hd0,0)
#         kernel /boot/vmlinuz-version ro root=/dev/sda1  console=tty0 console=
ttyS0,115200
#         initrd /boot/initrd-version.img
#boot=/dev/sda
default=1
timeout=5
#splashimage=(hd0,0)/boot/grub/splash.xpm.gz
#hiddenmenu
password --md5 $1$fUS9wQWB$.KWZno46Xc7iZipo6TJnY/
title Scientific Linux SLF (2.6.18-128.1.14.el5)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.18-128.1.14.el5 ro root=LABEL=/ console=tty0
console=ttyS0,115200
        initrd /boot/initrd-2.6.18-128.1.14.el5.img
```

Also make sure that /etc/inittab has the following entry as the last line:
co:12345:respawn:/sbin/agetty ttyS0 115200 vt100-nav
and /etc/securetty has the entry ttyS0.

Now you are ready to reboot and come up in the Xen kernel. Reboot the system..
Note that the RPMS say versions like xen-3.0.3, but in fact it is xen 3.1.2 back-ported into the red hat kernels. Alternatively, you could select the "virtualization" package during the system install, but that will give you a bunch of other programs too that you don't need**.**

Xen Configuration

The Xen daemons should be configured to start up automatically by

/sbin/chkconfig --add xend

/sbin/chkconfig --add xendomains

For our production system FermiGrid is using the method of Xen config files and "xm create" to configure and start the daemons. The Xen config files are found in /etc/xen. All of our High Availability machines are configured with two network interfaces, the public and the private, with
the latter being used for the Heartbeat system. To do that, there is a small modification that needs to be made.

In /etc/xen/scripts create the following script:

```
[root@fermigrid3 scripts]# cat my-network-bridge
#!/bin/sh
# start bridges on both eth0 and eth1
```

```
XENDIR="/etc/xen/scripts"

$XENDIR/network-bridge "$@" netdev=eth0 bridge=xenbr0 vifnum=0
$XENDIR/network-bridge "$@" netdev=eth1 bridge=xenbr1 vifnum=1

--------------------

modify /etc/xen/xend-config.sxp

< (network-script my-network-bridge)
---
> (network-script network-bridge)
```

Set up the Xen config files for each instance. Here are the non-commented lines from a sample instance. (Note that newer xen replaces sda1,sda2,sda3, with xvda1,xvda2,xvda3).

```
kernel = "/linux32/boot/vmlinuz-2.6-xen"
ramdisk = "/linux32/boot/initrd-2.6.18-xen.img"
memory = 2000
name = "fg3x1"
vif = [ 'mac=00:16:3e:05:03:01, bridge=xenbr0', 'mac=00:16:3e:05:03:0a,
bridge=xenbr1' ]
disk = [ 'phy:LG0/LV5,sda1,w','phy:LG0/LV10,sda2,w','phy:LG0/LV15,sda3,w' ]
netmask= "255.255.255.0"
gateway= "131.225.107.200"
hostname= "fg3x1.fnal.gov"
root = "/dev/sda1 ro"
extra = "4"
----------------------
```

Some notes on the conventions we use: All machines in the FermiGrid HA complex are using 00:16:3E:05: as the first four bytes of all their virtual MAC addresses. 00:16:3E is the stock MAC prefix that all Xen instances should be using. The Gratia Xen instances lead with 00:16:3E:04. The fifth byte is the number of the machine, i.e. all those on fermigrid5 will have 05, all on fermigrid6 will have 06, and so forth. The 6th byte is the number of the Xen instance itself. Thus fg5x1 is 00:16:3E:05:05:01. For eth1 the MAC address is shifted up by 8, for example 00:16:3E:05:05:09. We register all MACs with MISNET under the

base machine fermigrid5 for the base domain and all its Xens.

For disk partitioning, most of our Xen instances have three partitions exported to them and have the same fstab. The first partition is /, the second is swap, and the third is /usr/local The partitions are laid out using LVM.

Given a 10GB partition /dev/LG0/LV1 that will be the / partition for a xen instance, you do the following:
mount /dev/LG0/LV1 /mnt/xen1
cd /linux32
rsync -avDx * /mnt/xen1
cd /mnt/xen1
(can chroot if you want)
Change /etc/hosts to include the host name and IP of this xen instance.
Change /etc/sysconfig/network to include the new host name.
copy /etc/krb5.keytab from /usr/local/admin/keytab on fermigrid0
Copy the host certificates into /etc/grid-security, if you have any.
modify /etc/sysconfig/network-scripts/ifcfg-eth0 and /etc/sysconfig/network-scripts/ifcfg-eth1
to reflect the same IP's and MAC addresses that you have in the cfg file.
Change out of the /mnt/xen1 directory
umount /mnt/xen1
Importantfsck /dev/LG0/LV1
Now, ready to start up the xen instance

xm create xen1.cfg

You can watch the virtual console as the xen console comes up, and will often need to do so.
xm cons fg3x1
(use whatever the domain name is in the xen config file). We frequently see on the first boot of a Xen instance that /etc/sysconfig/network-scripts/ifcfg* files get moved to ifcfg*.bak files.
So you have to log in on the serial console (If you followed the steps above, the password will
automatically be the same as the root password of the base machine), and copy these files back,
and then restart the network.
.
You can list all the Xen domains running with "xm list".

Finally, to have certain Xen daemons start in boot, make a symlink in the /etc/xen/auto directory to the config file which is in the /etc/xen directory, and these daemons will start on boot automatically.

Xen Configuration, SLF5.2/Xen 3.1.2

Scientific Linux 5.2 and their cousins have good Xen management tools, virt-manager and virt-install. Given a partition, the virt-install command will make a virtual Xen machine, install the linux on it, and start it up. A sample virt-install command is as follows:

virt-install --name=fcdf0x1 --ram=6000 --vcpus=2 --mac=00:16:3E:0C:00:01 --os-type=linux --os-variant=rhel5 --location=http://linux1.fnal.gov/linux/slf52/x86_64/sites/Fermi --file=/dev/LG0/LV1 --extra-args="ks=http://131.225.107.31/kickstarts/fcdf0x1.cfg ip=131.225.240.49 netmask=255.255.255.0 gateway=131.225.240.200 dns=131.225.8.120"

By use of the virt-install you can create either a graphically-based unit and see its console using
virt-manager, or a non-graphically based unit with only a serial console. The newer systems we've deployed have been able to be deployed a lot faster this way. Note that the above example would create a partition table within /dev/LG0/LV1 logical volume. The individual partitions within that can be mounted from the dom0 when the machine is down by use of kpartx -a, which will create logical devices for each of the sub-partitions that can be mounted.

Circularly Replicated MySQL Installation & Configuration

This section describes the procedures to install a circular replicating, 2-node, highly available MySQL cluster. This system does not use MySQL Clustering engine (ndbd). Alternatively, this system is also described as a MySQL multimaster replication system.

This recipe is based on the Advanced MySQL Replication Techniques OnLamp article written by Giuseppe Maxia, How To Set Up Database Replication in MySQL by Falko Timme, and Chapter 15, Replication of the MySQL 5.0 Reference Manual. This guide is only valid for MySQL v5.0 and later.

Master Installation and Configuration (e.g., fg5x4.fnal.gov)

Install the server, client and rusers-server rpm packages:

```
yum -y install mysql mysql-server rusers-server
```

Edit /etc/my.cnf and add the following to the [mysqld] section.

```
###################
# For server tuning
set-variable = key_buffer_size=512M
set-variable = table_cache=512
```

```
set-variable = myisam_sort_buffer_size=100M
set-variable = max_connections=500
max_connect_errors=1000
# log=/var/log/mysqld.log
log-error=/var/log/mysqld.log
log-warnings=2

# If innodb is used.
innodb_flush_log_at_trx_commit=1
sync_binlog=1

# For replication. **Note** server-id and auto_increment_offset values!
server-id=1
log-bin=mysql-bin
auto_increment_increment=10
auto_increment_offset=1
master-host = fg6x4.fnal.gov
master-user = repl
master-password = <password>
relay-log=fg5x4-relay-bin
```

Make the `[mysql.server]` section look like this

```
user=mysql
basedir=/var/lib
log=/var/log/mysqld.log
log-error=/var/log/mysqld.log
log-warnings=2
Enable and start the server:
```

Make the `[mysqld_safe]` section look like this:

```
log=/var/log/mysqld.log
err-log=/var/log/mysqld.log
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Start the server:

```
chkconfig mysqld on
service mysqld start
```

Start the mysql client and issue these commands:

```
mysql> grant replication slave, replication client on *.*
        -> to 'repl'@'fg6x4.fnal.gov' identified by '<password>';
mysql> grant replication slave, replication client on *.*
        -> to 'repl'@'fg5x4.fnal.gov' identified by '<password>';
```

In the instance of a catastrophic single server or replication failure, start from here.

Log into the good mysql database server and start the client.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> show master status;
```

Record the values of File and Position. These will be used in the slave installation, later.

Remain logged into the mysql client to maintain the lock, then from another terminal do this:

```
cd /var/lib/mysql
tar -cvf /tmp/master-mysql-snapshot.tar \
--exclude=*relay-bin* \
--exclude=mysql-bin.* \
--exclude=*.info \
.

scp /tmp/master-mysql-snapshot.tar fg6x4:/var/tmp
```

Unlock the tables:

```
mysql> UNLOCK TABLES;
```

Slave Installation and Configuration (fg6x4.fnal.gov):

Install the server, client and rusers-server software:

```
yum install mysql mysql-server rusers-server
```

Untar the databases from the master:

```
cd /var/lib/mysql
tar -xvf /var/tmp/master-mysql-snapshot.tar
```

Edit /etc/my.cnf and add these lines to the [mysqld] section:

```
###################
# For server tuning
set-variable = key_buffer_size=512M
set-variable = table_cache=512
set-variable = myisam_sort_buffer_size=100M
set-variable = max_connections=500
max_connect_errors=1000
# log=/var/log/mysqld.log
log-error=/var/log/mysqld.log
log-warnings=2

# If innodb is used.
innodb_flush_log_at_trx_commit=1
sync_binlog=1

# For replication. **Note** server-id and auto_increment_offset values!
server-id=1
log-bin=mysql-bin
auto_increment_increment=10
auto_increment_offset=1
```

- master-host = fg5x4.fnal.gov
- master-user = repl
- master-password = <password>

```
relay-log=fg6x4-relay-bin
```

Make the `[mysql.server]` section look like this

```
user=mysql
basedir=/var/lib
log=/var/log/mysqld.log
```

```
log-error=/var/log/mysqld.log
log-warnings=2
```

Make the `[mysqld_safe]` section look like this:

```
log=/var/log/mysqld.log
err-log=/var/log/mysqld.log
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Enable and start the server:

```
chkconfig mysqld on
service mysqld start
```

Start the mysql client and issue these commands:

```
mysql> slave stop;
mysql> change master to
       -> master_log_file='<recorded log file name, above>',
       -> master_log_pos=<recorded log position, above>; # <- note
lack of quotes
       -> start slave;
```

To check the status of the slave on each machine:

```
mysql> show slave status\G;
```

THis command should yield an output similar to this:

```
*************************** 1. row ***************************
             Slave_IO_State: Waiting for master to send event
                Master_Host: fg5x4.fnal.gov
                Master_User: repl
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: mysql-bin.000001
        Read_Master_Log_Pos: 98
             Relay_Log_File: fg6x4-relay-bin.000002
              Relay_Log_Pos: 235
      Relay_Master_Log_File: mysql-bin.000001
          Slave_IO_Running: Yes
         Slave_SQL_Running: Yes
etc...
```

Create User For LVS Nanny Monitoring Script

```
    mysql> grant select on test.* to 'user1'@'fg5x0.fnal.gov' identified by
'<password>';
    mysql> grant select on test.* to 'user1'@'fg6x0.fnal.gov' identified by
'<password>';
```

Restart Procedure After an Unclean Shutdown

After the machines have been rebooted, and the mysql servers have been started, determine which mysql server is out-of-sync - log into both systems and issue the show slave status command. On the system that reports back waiting for master to reconnect perform the commands slave stop and slave start. Then re-issue the show slave status command to verify that the connection has been re-established.

Logrotate mysql.log

There is a bug in mysql logrotation using 'mysqladmin flush-logs': it doesn't work as it's supposed to. See http://bugs.mysql.com/bug.php?id=6061. A solution is to use the copytruncate option in the /etc/logrotate.d/mysql conf file:

```
/var/log/mysqld.log {
# create 600 mysql mysql
notifempty
daily
rotate 31
missingok
compress
# Because flush-logs doesn't work use this:
copytruncate
postrotate
# just if mysqld is really running
if test -x /usr/bin/mysqladmin && \
/usr/bin/mysqladmin ping &>/dev/null
then
/usr/bin/mysqladmin flush-logs
fi
endscript
}
```

Configure Iptables

Since only one host exists on the mysql server, we don't need to bring up a separate IP and we can use Horm's transparent proxy to rewrite IP packet headers. Therefore, use the mysql server iptables file.

Files and Directories to Sync Between fg5x4 and fg6x4

/etc/sysconfig/iptables

Mysql Upgrade Procedure

Open 3 terminals and log into fg5x0, fg5x4, and fg6x4.

On fg5x0 run the following command to monitor the lvsd processes. Specifically, watch for nanny to take the server offline during the upgrade:

```
tail -f /var/log/messages
```

You should see the following message appear 18 seconds after you shutdown the mysql server, below:

```
 Trouble. Recieved results are not what we expected from
(131.225.107.105)
```

On fg5x4, run the following:

```
cp /etc/my.cnf /etc/my.cnf.`date +%y%m%d`
```

```
/etc/init.d/mysqld stop
yum -y update mysql
/etc/init.d/mysqld start
 watch -n1 "mysql -B -e 'show slave status\G' | grep Slave_IO_State"
```

On fg5x0 you should see this appear on the terminal:

```
nanny[29408]: making 131.225.107.105:3306 available
```

After several second (up to 1 minute) the Slave_IO_State should change to:

```
Slave_IO_State: Waiting for master to send event
```

After seeing this displayed, you can Ctrl-C the "watch" command.

After the above messages are displayed on fg5x0 and fg5x4, perform the same set of procedures on fg6x4:

```
cp /etc/my.cnf /etc/my.cnf.`date +%y%m%d`
/etc/init.d/mysqld stop
yum -y update mysql
/etc/init.d/mysqld start
 watch -n1 "mysql -B -e 'show slave status\G' | grep Slave_IO_State"
```

On fg5x0 you should see this appear on the terminal:

```
nanny[29408]: making 131.225.107.109:3306 available
```

After several second (up to 1 minute) the Slave_IO_State should change to:

```
Slave_IO_State: Waiting for master to send event
```

Troubleshooting an Upgrade

If, after several minutes, the "waiting for master to send event" message is not displayed, start the mysql client and run 'show slave status\G' to determine what is failing. As a first attempt to re-establish the connection, run 'slave stop' and 'slave start'. If that fails to re-establish the connection, resort to reading Chapter 15, Replication of the MySQL 5.0 Reference Manual.

Piranha Linux Virtual Server Installation & Configuration

This section describes the steps to install and configure the Linux Virtual Server (LVS) using the Red Hat piranha package. It is meant to be a concise recipe, not a comprehensive HOWTO. The following table describes the virtual and real server names, as well as their IP addresses. Use this information to fill out the *Configure Virtual Server* section, below.

Virtual Server name	Virtual IP	Real Server 1 hostname	Real IP 1	Real Server 2 hostname	Re
LVS director and backup	NA	fg5x0.fnal.gov	131.225.107.146	fg6x0.fnal.gov	131.22
voms.fnal.gov	131.225.107.140	fg5x1.fnal.gov	131.225.107.102	fg6x1.fnal.gov	131.22
voms.opensciencegrid.org	131.225.107.138	fg5x1.fnal.gov	131.225.107.102	fg6x1.fnal.gov	131.22
gums.fnal.gov	131.225.107.139	fg5x2.fnal.gov	131.225.107.103	fg6x2.fnal.gov	131.22
saz.fnal.gov	131.225.107.126	fg5x3.fnal.gov	131.225.107.104	fg6x3.fnal.gov	131.22
fg-mysql.fnal.gov	131.225.107.124	fg5x4.fnal.gov	131.225.107.105	fg6x4.fnal.gov	131.22

LVS Installation

Generate and install host certificates. These will be used by the nanny monitoring scripts, described later.

On the master LVS node perform the following as root:

```
yum install piranha
yum install rusers
yum install mysql

chkconfig --on piranha-gui

service piranha-gui start

piranha-passwd
```

(and set the password)

Configure iptables

For transparent failover to a backup director, stateful connections should not be maintained in iptables. See the following:

http://www.austintek.com/LVS/LVS-HOWTO/HOWTO/LVS-HOWTO.failover.html#stateful_failover

"On failover, a director configured with no filter rules, can be replaced with an identically configured backup with no interuption of service to the client. There will be a time in the middle of the changeover where no packets are being transmitted (and possibly icmp packets are being generated), but in general once the new director is online, the connection between client and realserver should continue with no break in established tcp connections between the client and the realserver... If stateful filter rules are in place (e.g. only accept packets from ESTABLISHED connections) then after failover, the new director will be presented packets from tcp connections that are ESTABLISHED, but of which it has no record. The new director will REJECT/DROP these packets."

LVS director iptables

NOTE: for voms.fnal.gov and voms.opensciencegrid.org, the `--set-mark` value in iptables MUST match the `fwmark` in the lvs.cf file.

Restart IP tables:

```
service iptables restart
```

Configure Virtual Servers

Open a browser to http://fg5x0.fnal.gov:3636

The username is 'piranha', the password is whatever was set, above.

Click on "Global Settings" tab and fill in the fields:

Primary server public IP: 131.225.107.36
Primary server private IP: 192.168.18.16

Verify routing is: direct

IMPORTANT!!!! --> Click "Accept"

Click on the "Redundancy" tab and fill in the fields:

Redundant server public IP: 131.225.107.76
Redundant server private IP: 192.168.18.19

Heartbeat interval: 6
Assume dead after: 18
Heartbeat runs on port: 539
Monitor NIC links for failures: [X]

IMPORTANT!!!! --> Click "Accept"

Click on the "Virtual Servers" tab.

Click "Add"

Click the radio button next to the new unnamed service and click "Edit"

Fill in the fields:

Name: SERVICE_NAME:<port number>
Application port: <port number>
Protocol: tcp
Virtual IP Address: <IP of system associated with service>
Virtual IP Network Mask: 255.255.255.0
Firewall Mark: <port number> <- for convenience, only used for voms-admin services
Device: eth0:<N> <- where N is a unique number not held by another interface
Re-entry Time: 15
Service timeout: 6
Quiesce server: Yes
Load monitoring tool: rup
Scheduling: weighted least-connections
Persistence: (only used if using fwmark)

Persistence Network Mask: Unused

IMPORTANT!!!! --> Click "Accept"

Click on the "Real Server" tab

Click "Add"

Select the radio button next to the new, unnamed server and click "Edit"

Fill in the fields:

Name: <short name of the *first* real server hosting the service>
Address: <IP of this real server hosting the service>
Weight: <number of CPUS * speed of CPUs (in MHz) * speed of network (in Mbps)/1M>

IMPORTANT!!!! --> Click "Accept"

Click on the "Real Server" tab, again.

Click "Add"

Select the radio button next to the new, unnamed server and click "Edit"

Fill in the fields:

Name: <short name of the *second* real server hosting the service>
Address: <IP of this real server hosting the service>
Weight: <number of CPUS * speed of CPUs (in MHz) * speed of network (in Mbps)/1M>

IMPORTANT!!!! --> Click "Accept"

Click on the "Monitoring Scripts" tab.

Fill in the fields. If the service is an non-ssl enabled web server, the defaults are fine. If not, then either the path to a custom "Sending Program" will be needed or a custom "Expect" string is required.

The following list of scripts go into /usr/local/bin/ and the full path must be specified when calling the script. If the service is available, then the string "up" is issued and this is what the "Expect" field should contain.

lvs-ha-check-voms-admin.sh
lvs-ha-check-mysql.sh
lvs-ha-check-gums.sh

IMPORTANT!!!! Don't forget to add the "%h" to the end of the "Sending Program" field.

IMPORTANT!!!! --> Click "Accept"

After completing the above for each service, you should have a file that looks like lvs.cf.

At this point, restart the LVS service:

```
service pulse restart
```

Files and directories to sync between fg5x0 and fg6x0:

/etc/sysctl.conf
/etc/sysconfig/iptables
/etc/sysconfig/ha
/usr/local/bin

GUMS Installation & Configuration

This guide describes the steps to install and configure the Grid Users Management Server (GUMS) in conjunction with an LVS director. It is meant to be a concise recipe, not a comprehensive HOWTO.

GUMS Installation and Configuration

On fg5x2 and fg6x2, install the host and http certificates and keys.

Install rusers-server.

```
yum install rusers-server
```

Install http://software.grid.iu.edu/osg-1.2:gums per the instructions in the GUMS Installation Guide. Stop the tomcat server and disable the mysql server:

```
service tomcat-55 stop
service mysql stop
chkconfig mysql off
vdt-register-service --disable mysql
```

Dump the GUMS_1_3 database from the server, copy to a database server and reload:

```
mysqldump GUMS_1_1 > gums.sql
scp gums.sql fg5x4:/var/tmp
```

On fg5x4:

```
mysql < gums.sql
```

Add gums user on fg5x4:

```
mysql> grant all on GUMS_1_1.* to gums@fg5x2.fnal.gov identified by
```

```
'<password>';
mysql> grant all on GUMS_1_1.* to gums@fg6x2.fnal.gov identified by
'<password>';
```

Edit $VDT_LOCATION/vdt-app-data/gums/config/gums.config file to contact fg-mysql:

change: jdbc:mysql://fg-mysql.fnal.gov:3306/GUMS_1_1

Restart the tomcat server and enable gums-host-cron:

```
service tomcat-55 start
```

Edit $VDT_LOCATION/tomcat/v55/webapps/gums/WEB-INF/web.xml on fg5x2 to change
update time to 1 hr, and edit the same file on fg6x2 to change update time to 17 years. To make sure these changes aren't overwritten accidentally, change the file attribute to immutable on fg6x2:

```
if [ `hostname -s` -eq fg6x2 ]; then
    chattr -i $VDT_LOCATION/tomcat/v55/webapps/gums/WEB-INF/web.xml
fi
```

On each node, edit $VDT_LOCATION/tomcat/v55/webapps/gums/WEB-INF/config/gums.config.local and change the following line to point at the mysql server and port:

change: hibernate.connection.url="jdbc:mysql://fg-mysql.fnal.gov:3306/GUMS_1_1"

Create sym links in /var/log to the tomcat logs.

```
ln -s $VDT_LOCATION/tomcat/v55/logs /var/log/tomcat
```

Edit $VDT_LOCATION/tomcat/v55/webapps/gums/WEB-INF/classes/log4j.properties and change the logging parameters for the gums-service-admin portion for log4j.appender.adminFile.MaxFileSize=5000KB and log4j.appender.adminFile.MaxBackupIndex=1200. Other logging parameters can be changed as well.

Change permissions on /etc/grid-security/http/httpkey.pem and httpcert.pem to daemon.daemon.

```
chmod daemon.daemon /etc/grid-security/http/http{cert,key}.pem
```

Configure iptables

Since only one host exists on the gums server, we don't need to bring up separate IPs for each server and we can use Horm's transparent proxy to rewrite IP packet headers. Therefore, use this GUMS iptables section in the appendix.

Files and Directories to Sync Between fg5x2 and fg6x2

/etc/sysconfig/iptables
/etc/grid-security
/usr/local/vdt-2.0.0/
exclude
    $VDT_LOCATION/tomcat/v55/webapps/gums/WEB-INF/web.xml
    $VDT_LOCATION/tomcat/v55/logs
    $VDT_LOCATION/apache/log

Note--GUMS 1.3.15 which is distributed in the VDT 2.0.0 has got a way to distribute its gums configuration between redundant servers. FermiGrid is not yet using that

---

Troubleshooting

Note:

When using Horm's Transparent Proxy, the VIPs much match in the iptables on the LVS director and on the real servers.
For instance on the director (fg5x0) this line exists:

-A PREROUTING -d 131.225.107.124/32 -p tcp -m tcp --dport 3306 -j MARK --set-mark 3306

On the real servers the corresponding line is this:

-A PREROUTING -d 131.225.107.124 -p tcp --dport 3306 -j REDIRECT

Note:

On the director, in the iptables file the --set-mark value must match the value fwmark in the lvs.cf. For instance, using the same line from above:

-A PREROUTING -d 131.225.107.124/32 -p tcp -m tcp --dport 3306 -j MARK --set-mark 3306

In the lvs.cf file this value must be set:

    fwmark = 3306

Note:

On director and voms-admin real servers, enable access to port 8443 from outside fnal.gov, *but* only to the destination VIP (not all IPs).

Note:

Something is wrong with arptables, don't use it. Use arp_announce and arp_ignore. For more details, read the following:

http://www.ultramonkey.org/3/topologies/hc-ha-lb-eg.html

Note:

There is a bug in mysql logrotation using 'mysqladmin flush-logs': it doesn't work as it's supposed to. See http://bugs.mysql.com/bug.php?id=6061. A solution is to use the copytruncate option in the /etc/logrotate.d/mysql conf file:

```
 /var/log/mysqld.log {
        # create 600 mysql mysql
        notifempty
        daily
        rotate 31
        missingok
        compress
        # Because flush-logs doesn't work use this:
        copytruncate
    postrotate
        # just if mysqld is really running
        if test -x /usr/bin/mysqladmin && \
           /usr/bin/mysqladmin ping &>/dev/null
        then
           /usr/bin/mysqladmin flush-logs
        fi
    endscript
 }
```

Note:

On mysql servers, to clear the "too many failed connections" error message in /var/log/mysqld use 'mysqladmin flush-hosts' command on fg5x4 and fg6x4.

Appendix

---

iptables for mysql servers (click to download)

```
# mysql iptables
# horm's tranparent proxy for LVS
*nat
:PREROUTING ACCEPT [7:786]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
# The following IP must be the floating IP address from the LVS director
-A PREROUTING -d 131.225.107.124 -p tcp --dport 3306 -j REDIRECT
COMMIT

*filter
:INPUT ACCEPT [35:5488]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [22:1588]
```

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# Accept all from localhost
-A INPUT -s 127.0.0.1/255.255.255.255 -j ACCEPT

# Accept all from fg{5,6}x[1-4] to port 3306, *only*
-A INPUT -s 131.225.107.36/255.255.255.255 -p tcp -m tcp --dport 3306 -j
ACCEPT
-A INPUT -s 131.225.107.102/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.103/255.255.255.255 -p tcp -m tcp  --dport 3306
-j ACCEPT
-A INPUT -s 131.225.107.104/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.105/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.76/255.255.255.255 -p tcp -m tcp --dport 3306 -j
ACCEPT
-A INPUT -s 131.225.107.106/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.107/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.108/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT
-A INPUT -s 131.225.107.109/255.255.255.255 -p tcp -m tcp --dport 3306 -
j ACCEPT

# Accept all from within 131.225.0.0
-A INPUT -s 131.225.0.0/255.255.0.0 -j ACCEPT

# Allow ssh
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --dport 22 -j ACCEPT

# Allow kerberized telnet
-A INPUT -p tcp -m tcp --dport 23 -j ACCEPT
-A INPUT -p udp -m udp --dport 23 -j ACCEPT

# Allow klogin
-A INPUT -p tcp -m tcp --dport 543 -j ACCEPT
-A INPUT -p udp -m udp --dport 543 -j ACCEPT

# Allow kshell
-A INPUT -p tcp -m tcp --dport 544 -j ACCEPT
-A INPUT -p udp -m udp --dport 544 -j ACCEPT

# Allow eklogin
-A INPUT -p tcp -m tcp --dport 2105 -j ACCEPT
```

```
-A INPUT -p udp -m udp --dport 2105 -j ACCEPT

# drop everything else
-A INPUT -j DROP

COMMIT
```

---

iptables for LVS Director and Backup (click to download)

```
# LVS director iptables
*mangle
:PREROUTING ACCEPT [407:41667]
:INPUT ACCEPT [407:41667]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [278:22960]
:POSTROUTING ACCEPT [280:23232]

# For VOMS
-A PREROUTING -d 131.225.107.112/32 -p tcp -m tcp --dport 8443 -j MARK -
-set-mark 11280
-A PREROUTING -d 131.225.107.138/32 -p tcp -m tcp --dport 8443 -j MARK -
-set-mark 13880
COMMIT

*filter
:INPUT ACCEPT [35:5488]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [22:1588]

# Accept all from localhost
-A INPUT -s 127.0.0.1/255.255.255.255 -j ACCEPT

# Accept all from within 131.225.0.0
-A INPUT -s 131.225.0.0/255.255.0.0 -j ACCEPT

# Allow ssh
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --dport 22 -j ACCEPT

# Allow kerberized telnet
-A INPUT -p tcp -m tcp --dport 23 -j ACCEPT
-A INPUT -p udp -m udp --dport 23 -j ACCEPT

# Allow httpd
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p udp -m udp --dport 80 -j ACCEPT
```

```
# Allow VOMS ports
-A INPUT -p tcp -m tcp --dport 15001:15030 -j ACCEPT

# Allow https to voms.fnal.gov VOMS servers
-A INPUT -d 131.225.107.112 -p tcp -m tcp --dport 8443 -j ACCEPT
-A INPUT -d 131.225.107.112 -p udp -m udp --dport 8443 -j ACCEPT

# Allow https to voms.opensciencegrid.org VOMS servers
-A INPUT -d 131.225.107.138 -p tcp -m tcp --dport 8443 -j ACCEPT
-A INPUT -d 131.225.107.138 -p udp -m udp --dport 8443 -j ACCEPT

# Allow klogin
-A INPUT -p tcp -m tcp --dport 543 -j ACCEPT
-A INPUT -p udp -m udp --dport 543 -j ACCEPT

# Allow kshell
-A INPUT -p tcp -m tcp --dport 544 -j ACCEPT
-A INPUT -p udp -m udp --dport 544 -j ACCEPT

# Allow eklogin
-A INPUT -p tcp -m tcp --dport 2105 -j ACCEPT
-A INPUT -p udp -m udp --dport 2105 -j ACCEPT

# Allow MySQL
# -A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT
# -A INPUT -p udp -m udp --dport 3306 -j ACCEPT

# Allow VOMS ports
-A INPUT -p tcp -m tcp --dport 15001:15030 -j ACCEPT

# drop everything else
-A INPUT -j DROP
COMMIT
```

---

iptables for GUMS servers (click to download)

```
# iptables for GUMS real server
# horm's tranparent proxy for LVS
*nat
:PREROUTING ACCEPT [7:786]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -d 131.225.107.113 -p tcp --dport 8443 -j REDIRECT
COMMIT


*filter
```

```
:INPUT ACCEPT [35:5488]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [22:1588]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# Accept all from localhost
-A INPUT -s 127.0.0.1/255.255.255.255 -j ACCEPT

# Accept all from within 131.225.0.0
-A INPUT -s 131.225.0.0/255.255.0.0 -j ACCEPT

# Allow ssh
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --dport 22 -j ACCEPT

# Allow kerberized telnet
-A INPUT -p tcp -m tcp --dport 23 -j ACCEPT
-A INPUT -p udp -m udp --dport 23 -j ACCEPT

# Allow httpd
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p udp -m udp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8443 -j ACCEPT
-A INPUT -p udp -m udp --dport 8443 -j ACCEPT

# Allow klogin
-A INPUT -p tcp -m tcp --dport 543 -j ACCEPT
-A INPUT -p udp -m udp --dport 543 -j ACCEPT

# Allow kshell
-A INPUT -p tcp -m tcp --dport 544 -j ACCEPT
-A INPUT -p udp -m udp --dport 544 -j ACCEPT

# Allow eklogin
-A INPUT -p tcp -m tcp --dport 2105 -j ACCEPT
-A INPUT -p udp -m udp --dport 2105 -j ACCEPT

# Allow MySQL
-A INPUT -p tcp -m tcp --dport 3306 -j ACCEPT
-A INPUT -p udp -m udp --dport 3306 -j ACCEPT

# drop everything else
-A INPUT -j DROP
 COMMIT
```

LVS lvs.cf file gums only **(Click to download)**

```
serial_no = 119
primary = 131.225.107.36
primary_private = 192.168.18.16
service = lvs
backup_active = 1
backup = 131.225.107.76
backup_private = 192.168.18.19
heartbeat = 1
heartbeat_port = 539
keepalive = 6
deadtime = 18
network = direct
debug_level = NONE
monitor_links = 1
virtual MYSQL:3306 {
     active = 1
     address = 131.225.107.124 eth0:124
     vip_nmask = 255.255.255.0
     port = 3306
     expect = "up"
     use_regex = 0
     send_program = "/usr/local/bin/lvs-ha-check-mysql.sh %h"
     load_monitor = rup
     scheduler = wlc
     protocol = tcp
     timeout = 6
     reentry = 15
     quiesce_server = 1
     server fg5x4 {
         address = 131.225.107.105
         active = 1
         weight = 3
     }
     server fg6x4 {
         address = 131.225.107.109
         active = 1
         weight = 3
     }
}

virtual gums-fg5x2:8443 {
     active = 1
     address = 131.225.107.113 eth0:113
     vip_nmask = 255.255.255.0
     port = 8443
     expect = "up"
```

```
    use_regex = 0
    send_program = "/usr/local/bin/lvs-ha-check-gums.sh %h"
    load_monitor = rup
    scheduler = wlc
    protocol = tcp
    timeout = 6
    reentry = 15
    quiesce_server = 1
    server fg5x2.fnal.gov {
        address = 131.225.107.103
        active = 1
        weight = 3
    }
    server fg6x2.fnal.gov {
        address = 131.225.107.107
        active = 1
        weight = 3
    }
}
```

LVS Check Mysql script (click to download)

```
    #!/bin/bash
    #lvs_ha_check_mysql.sh



    if [ $# -eq 0 ]; then
            echo "host not specified"
            exit 1
    fi

    /usr/bin/mysql test -h $1 -u user1 --password=pass1 -B -e 'show tables;'
    &> /dev/nu
    ll

    if [ $? -eq 0 ]; then
            echo "up"
    else
            echo "down"
    fi
```

LVS Check GUMS script (click to download)

```
#!/bin/bash
# lvs-ha-check-gums.sh

if [ $# -eq 0 ]; then
        echo "host not specified"
        exit 1
fi

curl -s --insecure --cert /etc/grid-security/hostcert.pem --key /etc/
grid-security/hostkey.pem https://${1}:8443/gums/services/
GUMSAuthorizationServicePort | grep "Hi there" &> /dev/null

if [ $? -eq 0 ]; then
        echo "up"
else
        echo "down"
fi
```