High Availability Deep Dive

Welcome to our Percona Live MySQL Conference & Expo 2012 tutorial! This session includes highly interactive content where you will be able to follow along setting up and running your very own high availability cluster. Since our time is extremely limited, please follow the instructions below to set up your machine before we start. It should only take about 5 minutes of your time.

Virtual Environment Setup Instructions

Setup Instructions for libvirt / Qemu / KVM

If your laptop runs Linux, then this is the preferred option. Please note that we're *only* supporting Qemu/KVM *with libvirt* for this session. We are not running standalone Qemu or KVM.

Prerequisites

- Please make sure your laptop has both the most recent version of libvirt available for your distribution.
- If your machine comes with CPU extensions for hardware virtualization (Intel VT or AMD SVM), please *turn them on* in your machine BIOS. Most vendors ship with hardware virtualization disabled by default.

On Debian/Ubuntu systems, you can use the kvm-ok command to quickly check whether your box supports hardware virtualization, and whether it is enabled.

- Install the latest version of Qemu and/or KVM available for your distribution. In recent distributions this may well be a single package, named qemu-kvm or similar, that supports both standalone Qemu and KVM.

Once your system meets all prerequisites, please continue with the next step.

Network installation

Please import the virtual network definitions into your libvirt configuration, and enable them.

- Copy the `pacemaker.xml` and `drbd.xml` files found in `qemu/networks` to the appropriate location for your distro. Normally, this `should` be `/etc/libvirt/qemu/networks`.

Please make sure that the newly defined networks (10.0.2.0/24 and 172.30.222.0/24) do not clash with existing network definitions. If that is the case, please see one of us to resolve the issue.

- Import both networks into your libvirt configuration with `virsh net-define <path to xml file>`.
- After the import, enable both networks with `virsh net-start pacemaker` and `virsh net-start drbd`.
- You may optionally auto-start the networks on libvirtd startup. To do so, issue `virsh net-autostart pacemaker` and `virsh net-autostart drbd`. You are welcome to skip this step, however please remember to start both networks prior to starting any of the two virtual machines.

Virtual machine installation

Once the virtual networks are properly set up, you can continue importing the domain (virtual machine) definitions.

First, install both `.vmdk` files into the appropriate location for virtual machine images (normally `/var/lib/libvirt/images`).

- If you are using KVM, please copy both XML files from the `kvm` directory into the appropriate location for your distribution (normally `/etc/libvirt/qemu`).
- If you are using Qemu, please copy both XML files from the `qemu` directory into the appropriate location for your distribution (normally `/etc/libvirt/qemu`).

libvirt stores KVM and Qemu domain definitions in the same path — there is no separate `/etc/libvirt/kvm` directory.

- Import both domains into your libvirt configuration with `virsh define <path to xml`

file>.
- After the import, start both networks with `virsh start pacemaker-1` and `virsh start pacemaker-2`.

Connecting to the virtual machines

There are three different ways you may connect to the virtual machines:
- Secure Shell. This is the recommended method. You can shell in as `root` to 172.30.222.1 (pacemaker-1) and 172.30.222.2 (pacemaker-2). The `root` login password is `pacemaker`. You may want to install your personal SSH public key in `/root/.ssh/authorized_keys` in order to enable passwordless logins.
- Virtual serial console. You can log in to the domains with `virsh console <domainname>`, where `<domainname>` is either `pacemaker-1` or `pacemaker-2`.
- Virtual manager console. If you have `virt-manager` installed on your machine, you can log in using the VNC virtual console. Simply open `virt-manager --connect=qemu:///system` and double-click one of the running virtual machines to open a console window.

X is not installed in the virtual machines, in order to to keep them small and lean.

Testing

Once you are logged into the machines, please perform the following test to ensure proper network connectivity:
- While logged into `pacemaker-1`, please ping `pacemaker-2` at 172.30.222.2
- While logged into `pacemaker-2`, please ping `pacemaker-1` at 172.30.222.1

Setup Instructions for VMware

This is the an option if your machine runs Windows or Mac OS X.

VMware Server and Workstation both have a pretty good track record with running our images. VMware Player, however, occasionally produces odd issues, some of which we've previously tried to resolve only to get our noses bloody. Thus, if you're running into a setup issue on Player, please install one of the other free VMware products and see if that resolves the problem.

Prerequisites

- Please make sure your VMware product installed at least one "NAT" and one "Host-only" virtual network during installation.
- Set the "NAT" network to use the 10.0.2.0/24 subnet.
- Configure your physical box to use the 10.0.2.2 address in that virtual network.

Virtual machine installation

Please copy all `.vmdk` files, plus the files from the vmware directory (4 files in total) to a convenient location on your machine. All files must be copied to the same directory; do not split them up into separate folders. Then, simply click on both `.vmx` files to import the virtual machine definitions.

Connecting to the virtual machines

There are three different ways you may connect to the virtual machines:
- Secure Shell. This is the recommended method. You can shell in as `root` to 172.30.222.1 (`pacemaker-1`) and 172.30.222.2 (`pacemaker-2`). The `root` login password is `pacemaker`. You may want to install your personal SSH public key in `/root/.ssh/authorized_keys` in order to enable passwordless logins.

- VMware console. You will be able to connect to the `tty0` terminal on both virtual machines through the VMware Server/Workstation/Player console.

X is not installed in the virtual machines, in order to to keep them small and lean.

Testing

Once you are logged into the machines, please perform the following test to ensure proper network connectivity:

- While logged into `pacemaker-1`, please ping `pacemaker-2` at 172.30.222.2
- While logged into `pacemaker-2`, please ping `pacemaker-1` at 172.30.222.1

Setup Instructions for VirtualBox

This is the preferred option if your machine runs Windows or Mac OS X.

Prerequisites

- Install Virtualbox from the files on the USB stick

Virtual machine installation

- start VirtualBox
- in the file menu, click on “Import Appliance...”
- Select the pacemaker-1.ova file and import it
- Do the same for pacemaker-2.ova
- Click to highlight “Pacemaker-1” in click on “start”
- Click to highlight “Pacemaker-1” in click on “start”

Connecting to the virtual machines

There are two different ways you may connect to the virtual machines:

- Secure Shell. This is the recommended method. You can login as *root* with the password *pacemaker*. Use IP 127.0.0.1 port 22222 for access to pacemaker-1 and port 22223 for pacemaker-2.
- You can use the VirtualBox window and access directly. The windows will grab the mouse, to release it, press the right side “ctrl” key.

Testing

Once you are logged into the machines, please perform the following test to ensure proper network connectivity:

- While logged into pacemaker-1, please ping pacemaker-2 at 172.30.222.2
- While logged into pacemaker-2, please ping pacemaker-1 at 172.30.222.1

Configuring MySQL High Availability in Pacemaker with DRBD

In this part of our tutorial, you will learn how to configure and set up a MySQL high availability cluster using the Corosync cluster messaging layer, the Pacemaker cluster resource manager, MySQL, and DRBD for storage replication.

Configuring the DRBD Block-Level Storage Replication Facility

To configure DRBD, you'll normally have to add a resource definition in the /etc/drbd.d directory. We've completed this task for you, it's in /etc/drbd.d/mysql.res. All we have left to do is initialize this DRBD resource.

First, we create DRBD's metadata. This step must be completed on both cluster nodes:

- `drbdadm create-md mysql`

Then, we need to bring the resources up for the first time. Later on, our cluster manager will take care of this for us, but for now, we need to do the following steps manually (again, on both nodes):

- `modprobe drbd`
- `drbdadm up mysql`

And finally, we need to kick off the initial full synchronization (on **only one node** this time):

- `drbdadm -- -o primary mysql`

You can check the DRBD synchronization status by taking a look at the /proc/drbd virtual file:

- `cat /proc/drbd`

At this point our DRBD resource is fully usable, and we can create a filesystem on it (again, on only one node – the same that you kicked off the full synchronization on:

- `mkfs -t ext3 /dev/drbd/by-res/mysql`

Configuring the Corosync Cluster Messaging Layer

Corosync's central configuration file is `/etc/corosync/corosync.conf`, which we have preconfigured for you. The one modification you need to make is to set the `bindnetaddr` parameter to match your virtual environment's configuration. `bindnetaddr` should match the network address of the `eth1` network interface.

Once you have made that modification, and synchronized `/etc/corosync/corosync.conf` across both nodes, you may start the Corosync daemon with one of the following commands, as `root`:

- `/etc/init.d/corosync start` *or*
- `service corosync start`

When the Corosync daemon is running on both nodes, you may verify the health of your cluster membership with the following commands:

- `corosync-cfgtool -s` *and*
- `corosync-objctl | grep member`

Both of your corosync rings should display as being healthy, and the member list should contain 3 entries each for both nodes.

Configuring the Pacemaker Cluster Resource Management Daemon

Running Pacemaker on top of Corosync requires two things:

- Adding the `pacemaker` service in `/etc/corosync/service.d` (we have prepared this for you);
- Starting the `pacemakerd` daemon.

You can start Pacemaker with one of the following commands:

- `/etc/init.d/pacemaker start` *or*
- `service pacemaker start`

When `pacemakerd` starts, it spawns a number of child processes, such as `pengine`, `crmd`, `cib` and `lrmd`. Check with `ps -AHfww`.

Once you have verified the Pacemaker processes are running, you may check your cluster health with the `crm_mon` command. Both nodes should display as `Online`, and no resources should be configured.

At this point, you have a working cluster configuration. All that is left to do is fill it with cluster resources.

Setting Cluster Properties

In your `/root` directory, you will find a set of file names ending in `.crm`. These are Pacemaker configuration snippets to be parsed and imported by the Pacemaker crm shell.

The first snippet we'll import is one that sets the `no-quorum-policy` and `stonith-enabled` cluster properties. Import it with:

- `crm configure load update /root/properties.crm`

Configuring a Highly Available Cluster IP Address

The first simple cluster resource we'll configure is a floating IP address, also known as a Virtual IP address or VIP. To do so, load the `ip.crm` snippet into your configuration:

- `crm configure load update /root/ip.crm`

The IP address that this configures is 172.33.222.100. It will be assigned to one of your cluster nodes. Verify the availability of your VIP with the following commands:

- `crm_mon`
- `ip address show dev eth0`

You can also simulate resource failure with this command which removes the VIP:

- `ip address delete 10.0.2.100/24 dev eth0`

Pacemaker will now automatically recover this resource. Confirm with the following command, which should not only display the resource being recovered, but subsequently also showing its fail count:

- `crm_mon -rf`

When you're done, you may clean up the resource, i.e. remove its fail count:

- `crm resource cleanup p_ip_mysql`
- `crm_mon -rf`

Adding a DRBD-backed Filesystem to the Cluster Configuration

With the following step, you will put the DRBD resource, and the filesystem which it hosts, under Pacemaker management. To do so, we'll set up a DRBD master/slave set, a Filesystem primitive resource, and a couple of constraints. All of these are in the next configuration snippet you'll load,

drbd.crm.

- `crm configure load update fs.crm`
- `crm_mon -rf`

At this point, Pacemaker will *promote* the DRBD resource to the `Primary` role on one of the nodes, and mount the DRBD device to `/var/lib/mysql` on the same node.

You can verify this with:

- `cat /proc/drbd`
- `mount`
- `drbd-overview`

Adding MySQL

Finally, we'll now add MySQL. First, we need to bootstrap a MySQL database. On the node that currently has your DRBD device in the Primary role and mounted to /var/lib/mysql, run the following command:

- `mysql_install_db --user=mysql`

Now, we can add the database to the cluster configuration. At the same time, we can put the DRBD/ filesystem combination, the IP address, and MySQL itself under one resource group:

- `crm configure load update mysql.crm`

Note how the crm shell, when you load this configuration snippet, automatically updates existing constraints to now apply to the `g_mysql` group, rather than the `p_fs_mysql` primitive.

Again, you may check on your cluster's current status with the following commands:

- `cat /proc/drbd`
- `mount`
- `drbd-overview`
- `ps -AHfww`
- `netstat -ntp`
- `crm_mon -rf`

Notes on MySQL configuration and requirements for HA

MyISAM

MyISAM is not durable and recovery or a MyISAM table after a crash may take a long time or simply fail. This is why, any table with non-trivial write load must not be using the MyISAM storage engine.

log-bin

If the HA MySQL/Pacemaker/DRBD pair is designed to become a replication master, the log-bin variable must be set to a value. This will cause the binary log files to be named according to the value set instead of the host name so that slaves will easily reconnect after a failover.

sync_binlog

If the HA MySQL/Pacemaker/DRBD pair is designed to become a replication master, sync_binlog should be set to avoid losing transaction in case of a crash.

bind

The solution relies on virtual IP (VIP) so the bind option, if used, must be set to the virtual IP. It is not mandatory to set it since by default, it binds on all IPs, including the VIP.

innodb_log_file_size

This parameter basically tune the recovery time, the smaller the innodb_log_file_size, the faster the recovery. Since this parameter is directly linked to the amount of dirty pages that are allowed in the buffer pool, small values causes poor write performance. With recent versions of MySQL and Percona server, 512 MB will recover in about 1min. Old version may be up to 20x slower. Best is to experiment and adjust with your write load and desired failover time.

innodb_auto_lru_dump / innodb_buffer_pool_restore_at_startup

Percona server has a very handy feature for a MySQL/Pacemaker/DRBD type setup, it is the ability to save the list of pages included in the buffer pool and reload these at startup. This feature reduce the period of time after a failover when the performance is degraded because the buffer pool is not warm.

Configuring MySQL High Availability in Pacemaker with Replication

In this part we will setup another of MySQL HA solution, this time using replication.

Cleanup of the Pacemaker configuration

Since the Pacemaker configuration we will use for this setup is quite different from the DRBD setup, it is easier to start from scratch. Resources must be stopped before erasing the configuring. Setting both nodes to standby is an easy way to achieve this.

- `crm node standby pacemaker-1`
- `crm node standby pacemaker-2`
- `crm configure erase`

Setup MySQL for replication

In order to setup replication, we need to have an exact copy of the database on both hosts. Since we just went through the Pacemaker/DRBD part, why not cheat a bit and reuse the database on the DRBD device.

- `/etc/init.d/drbd stop`
- `rm /etc/drbd.d/MySQL.res`
- `cat /root/fstab.part >> /etc/fstab`
- `mount /var/lib/mysql`

At this point we can start MySQL on both servers

- `/etc/init.d/mysql start`

and the set the grants for the replication solution:

- `mysql < /root/grants.sql`
- `/etc/init.d/mysql stop`

Reload the cluster properties

As we did previously, we need to set the `no-quorum-policy` and `stonith-enabled` cluster properties. Import these with:

- `crm configure load update /root/properties.crm`

Load the node definition

The replication setup requires that you define an IP attribute per node to be used for replication in the “Change master to” commands. To load the IP attributes, run:

- `crm configure load update /root/IP_attributes.crm`
- `crm node show`

Load the VIP primitives

The solution will use one VIP the *writer_vip* for the write operations and 2 VIPs dedicated for read operations. The following example uses 2 distinct IPAddr2 resources for the reader VIPs but a clone set could also have been used.

- `crm configure load update /root/VIP_repl.crm`
- `crm configure show`

Load the MySQL primitive

Now we can load the MySQL primitive. Compared to the previous usage, you'll note many additional parameters.

- `crm configure load update /root/mysql_repl.crm`
- `crm configure show`

Load the master-slave clone set

Pacemaker has the ability to handle resources in different states, master and slave. In order for the solution to work properly, do not forget to enable notification.

- `crm configure load update /root/ms.crm`
- `crm configure show`
- `crm status`

Load the location rules for the VIPs

The solution requires to be able to manage the reader VIP based on the state of the replication. This is handle by transient node attribute *readable* and the following location rules. An optional colocation rule is also added to spread the reader VIPs. On large cluster the score colocation rules could be configured to avoid a meltdown of cluster with all reader VIPs moving to master under high load.

For the writer VIP, we just colocate it with the master role and order it to start only after the promotion of the master.

- `crm configure load update /root/readervip_rules.crm`
- `crm configure load update /root/writervip_rules.crm`
- `crm configure show`
- `crm status`

Feedback

If you have any feedback for this guide, we're always eager to hear it. Our email addresses are florian@hastexo.com and yves@percona.com.