17
May
2012

An Infrastructure for Server Clusters for High Availability

As announced in our Cluster Services Built With FOSS post, LinuxForce's Cluster Services are built exclusively with Free and Open Source Software (FOSS). Here is an expanded outline of the basic architecture of our approach to High-Availability (HA) clustering.

Overview

In any HA deployment there are two main components: hosts and guests. The hosts are the systems which are the core of the cluster itself. The host runs with very limited services dedicated for the use and functioning of the cluster. The host systems handle resource allocation, from persistent storage to RAM to the number of CPUs each guest gets. The host machines give an "outside" look at guest performance and give the opportunity to manipulate them from outside the guest operating system. This offers significant advantages when there are boot or other failures which traditionally would require physical (or at least console) access to debug. The guests in this infrastructure are the virtual machines (VMs) which will be running the public-facing services.

On the host, we define a number of "resources" to manage the guest systems. Resources are defined for ping checking the hosts, bringing up shared storage or storage replication (like drbd) as primary on one machine or the other and launching the VMs.

In the simplest case, the cluster infrastructure is used for new server deployments, in which case the operating system installs are fresh and the services are new. More likely a migration from an existing infrastructure will be necessary. Migrations from a variety of sources are possible including from physical hardware, other virtualization technologies (like Xen) or different KVM infrastructures which may already use many of the same core features, like shared storage. When a migration is required downtime can be kept to a minimum through several techniques.

Hardware configuration

The first consideration when you begin to build a cluster is the hardware. The basic requirement for a small cluster is 3 servers and a fast dedicated network backplane to connect the servers. The three servers can all be active as hosts, but we typically have a configuration where two machines are the hosts and a third, less powerful arbitrator system is available to make sure there is a way to break ties when there is resource confusion.

Two live resource hosts

These systems will be where the guests are run. They should be as similar as possible down to the selection of processor brand and amount of RAM and storage capabilities so that both machines are capable of fully taking over for the other in case of a failure, thus ensuring high availability.

The amount of resources required will be heavily dependent upon the services you're running. When planning we recommend thinking about each guest as a physical machine and how many resources it needs, allowing room for inevitable expansion of services over time. You can over-commit both CPU and RAM on KVM, so you will want to read a best practices guide such as

Chapter 6: Overcommitting with KVM. Disk space requirements and configuration will vary greatly depending upon your deployment, including the ability to use shared storage backplanes and replicated RAID arrays, but Linux Software RAID will typically be used for the core operating system install controlling each physical server. Additionally, using a thorough testing process so you know how your services will behave if they run out of resources is critical to any infrastructure change.

Tie-breaker (arbitrator)

A third server is required to complete quorum for the cluster. In our configuration this machine doesn't need to have high specs or a lot of storage space. We typically use at least RAID1 so we have file system redundancy for this host.

1000M switch

A fast switch whose only job is to handle traffic between the three machines is highly recommended for assured speed of these two vital resources:

1. Storage backplane
2. Corosync/Pacemaker communication

It's best to keep these off a shared network, which may be prone to congestion or failure, since fast speeds for both these resources are important for a properly functioning cluster.

Key software components

There are many options when it comes to selecting your HA stack, from which Linux distribution to use, to what storage replication system to use. We have selected the following:

Debian GNU/Linux

Like most LinuxForce solutions, we start with a base of Debian stable, currently Debian "Squeeze" 6.0. All of the software mentioned in this article comes from the standard Debian stable repository and is open source and completely free of charge.

Logical Volume Manager (LVM)

We use LVM extensively throughout our deployments for the flexibility of easy reallocation of filesystem resources. In a cluster infrastructure it is used to create separate disk images for each guest and then may be used again inside this disk image for partitioning.

Distributed Replicated Block Device (DRBD)

DRBD is used for replicating storage between the two hosts which have their own storage. Storage needs could also be met by shared storage or other data replication mechanisms.

Kernel-based Virtual Machine (KVM)

Since hardware-based virtualization is now ubiquitous on modern server hardware we use KVM for our virtualization technology. It allows fully virtualized VMs running their own unmodified kernels to run directly on the hardware without the overhead of a hypervisor or emulation.

Pacemaker & Corosync

Pacemaker and Corosync are used together to do the heavy lifting of the cluster management. The two services are deeply intertwined, but at the core Pacemaker handles core configuration of the resources themselves, and Corosync handles quorum and "aliveness" checks of the hosts and resources and determination of where resources should go.

Conclusion

We have deployed this infrastructure for mission critical services including DNS, FTP and web server infrastructures serving everything from internal ticketing systems to high-traffic public-facing websites. For a specific example of implementation of this infrastructure, see Laird Hariu's report File Servers – The Business Case for High Availability where he covers the benefits of HA for file servers.

Posted by Elizabeth Krumbach *in* Development, Systems Management, Tech Notes, Virtualization, *0 comments*