17
Jul
2012

5 things about FOSS Linux virtualization you may not know

In January I attended the 10th annual Southern California Linux Expo. In addition to speaking and running the Ubuntu booth, I had an opportunity to talk to other sysadmins about everything from selection of distribution to the latest in configuration management tools and virtualization technology.

I ended up in a conversation with a fellow sysadmin who was using a proprietary virtualization technology on Red Hat Enterprise Linux. Not only did he have surprising misconceptions about the FOSS (Free and Open Source Software) virtualization tools available, he assumed that some of the features he was paying extra for (or not, as the case may be) wouldn’t be in the FOSS versions of the software available.

Here are five features that you might be surprised to find included in the space of FOSS virtualization tools:

1. Data replication with verification for storage in server clusters

When you consider storage for a cluster there are several things to keep in mind:

- Storage is part of your cluster too, you want it to be redundant
- For immediate failover, you need replication between your storage devices
- For data integrity, you want a verification mechanism to confirm the replication is working

Regardless of what you use for storage (a single hard drive, a RAID array, or an iSCSI device), the open source DRBD (Distributed Replicated Block Device) offers quick replication over a network backplane and verification tools you can run at regular intervals to ensure deta integrity.

Looking to the future, the FOSS distributed object store and file system Ceph is showing great promise for more extensive data replication.

2. Automatic failover in cluster configurations

Whether you’re using KVM Kernel-based Virtual Machine or Xen, automatic failover can be handled via a couple of closely integrated FOSS tools, Pacemaker and Corosync. At the core, Pacemaker handles core configuration of the resources themselves and Corosync handles quorum and “aliveness” checks of the hosts and resources and logic to manage moving Virtual Machines.

3. Graphical interface for administration

While development of graphical interfaces for administration is an active area, many of the basic tasks (and increasingly, more complicated ones) can be made available through the Virtual Machine Manager application. This manager uses the libvirt toolkit, which can also be used to build custom interfaces for management.

The KVM website has a list of other management tools, ranging from command-line (CLI) to Web-based: www.linux-kvm.org/page/Management_Tools

As does the Xen wiki: wiki.xen.org/wiki/Xen_Management_Tools

4. Live migrations to other hosts

In virtualized environments it's common to reboot a virtual machine to move it from one host to another, but when shared storage is used it is also possible to do *live* migrations on KVM and Xen. During these live migrations, the virtual machine retains state as it moves between the physical machines. Since there is no reboot, connections stay intact and sessions and services continue to run with only a short blip of unavailability during the switch over.

Documentation for KVM, including hardware and software requirements for such support, can be found here: www.linux-kvm.org/page/Migration

5. Over-allocating shared hardware

KVM has the option to take full advantage of hardware resources by over-allocating both RAM (with adequate swap space available) and CPU. Details about over-allocation and key warnings can be found here: Overcommitting with KVM.

Conclusion

Data replication with verification for storage, automatic failover, graphical interface for administration, live migrations and over-allocating shared hardware are currently available with the FOSS virtualization tools included in many modern Linux distributions. As with all moves to a more virtualized environment, deployments require diligent testing procedures and configuration but there are many on-line resources available and the friendly folks at LinuxForce to help.

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