oVirt Node

November 1, 2011

Mike Burns
Alan Pevec
Perry Myers
Agenda

- Introduction
- Architecture Overview
- Deployment Modes
- Installation and Configuration
- Upgrading
- Configuration Persistence
- Future Features
- Discussion
Introduction to oVirt Node

- Dedicated Hypervisor
  - Minimum OS needed to run and manage VMs
  - Well defined management interfaces and APIs
- Small Footprint
  - Less than 150MB image size
  - ~850MB disk space required for installation
  - 512MB runtime RAM footprint
- Built from Fedora components
  - Supports same hardware as Fedora
- Utilizes KVM
- Includes VDSM for VM Management
Architecture – RPMs

- ovirt-node
  - A TUI interface for installation and configuration
  - Automatic installation and configuration
  - Modifications to sysinit scripts

- ovirt-node-tools
  - A set of kickstart files
  - Minimal package list
  - Blacklisting for image minimization
  - Used to create an image
Architecture - Image

- Built using kickstarts from ovirt-node-tools package
- Provided in ISO format
- Installs to local hard drive, flash drive, SD card
- Minimal state information persisted to a config partition
- Multiple upgrade paths
  - Booting a new image
  - In place upgrade
livecd-tools

- Utility for create an ISO image containing a LiveOS
  - http://fedoraproject.org/wiki/FedoraLiveCD
- Uses the following as input
  - Kickstart file to automate OS installation
    - package list (RPMs)
    - custom %post script for configuration
    - scripts for minimizing image size using file blacklisting and forced package removal
  - Yum Repositories for OS packages
  - Output is an ISO image
Key Packages

- qemu-kvm – provides KVM virtualization platform
- qemu-kvm-tools – kvmtrace and kvm_stat for debugging utilities
- vdsm – daemon for managing the node from oVirt Engine
- vdsm-cli – command line interface to VDSM daemon
- libvirt – virtualization API and VM control daemon
- spice-server – Provides guest remote connections
Manifest Files

- The produced image file contains manifest files in the top-level isolinux folder
- Manifests document the content of the final image after minimization (blacklisting)
  - manifest-dir.txt.bz2 – directories in the image
  - manifest-file.txt.bz2 – files in the image
  - manifest-license.txt – licenses for all installed RPMs
  - manifest-owns.txt.bz2 – file ownership by RPM
  - manifest-rpm.txt – installed RPMs
  - manifest-srpm.txt – source RPMs for installed RPMs
Deployment Modes

- **CD-ROM**
  - Burn ISO image to writable CD with standard tools
  - Boot server from CD and install to local hard disk
- **Flash Memory (USB stick or SD card)**
  - Use livecd-iso-to-disk to copy image to USB or SD card
  - Boot from USB/SD and install to local hard disk
- **Network (PXE) Boot**
  - Use livecd-iso-to-pxeboot on the iso
  - Deploy generated vmliuz0/initrd0 files to PXE/tftp server
  - Boot server from PXE
Installation

- The oVirt Node image currently needs to be installed to a hard disk or flash drive to run
- After installation, boot method should be changed to hard disk
- There are two modes of installation
  - Booting to installation TUI
  - Autostart via kernel command line arguments
- Booting from CD or Flash memory will bring you to a Boot Menu
Boot Menu

Note: *linux rescue can be passed to the boot: parameter to boot in rescue mode*
This screen will warn if Hardware Virtualization is not enabled on the host (see bottom on screen shot above)
TUI Installation – Disk Selection

- Select a device to boot from (left screen)
- Select 1 or more disks for storing configuration data and swap (right screen)
TUI Installation – Admin Password

- Set a password for the admin user
- Proceeding from this screen starts the installation
TUI Installation -- Finishing

- Installing Bootloader Configuration On /dev/vda
- 75%

- oVirt Node Hypervisor Installation Finished Successfully
  - <Reboot>
### Configuration - Status

- Get overall status of the system
- See number of VMs running
Configuration – Networking

- Choose device to setup for management bridge
- Supports vlans
Enable password based ssh authentication
Reset admin password
Configuration - Logging

- Configure Remote logging server
Configuration - kdump

- Configure kdump server
Configuration – Remote Storage

- Setup an iSCSI Initiator Name
- One is randomly generated during installation
• Register to the oVirt Engine management server
Configuration – Monitoring

- Connect with a remote collectd server
Automatic Installation and Configuration

• All configuration in the TUI can be automated with kernel command line parameters
• Ideal for PXE boot environments
• Requires storage_init and BOOTIF parameters
• adminpw parameter recommended for management after installation
Upgrading

- oVirt Node image is a dedicated appliance
  - no yum/rpm upgrading in the live image
  - Rootfs is non-persistent so upgrades are lost
    - Warning: Runtime rootfs (/) is in-memory overlay. Writing excessive amounts of data to it can cause out of space issues
    - It is mounted read-only by default to avoid this issue

- Three upgrade paths supported
  - Update the PXE server and set host to network boot
  - Boot from new media (CD, USB, SD)
  - In-place upgrade

- ISO/USB/PXE upgrades must specify upgrade on the kernel command line to trigger upgrade logic
Configuration Persistence

- Root FS is mounted read-only
  - even if remounted RW, changes are not persisted
- Current persistence uses rc.sysinit stateless support
- Important files are persisted automatically by oVirt and VDSM as needed
- To manually persist a file, use the persist command:
  - # persist /etc/hosts
- The /config partition is only a few MB by default, so use sparingly.
Roadmap – Stateless

• Feature
  • Be able to boot from media/PXE and fetch configuration from remote storage server
  • No need for local storage
    • Except for swap if running with overcommit
  • Post-Boot, find the configuration server (DNS SRV or similar) and download configuration bundle
  • Apply configuration bundle and report to oVirt Engine
Roadmap - Plugins

- Ability to add software (drivers, configuration, monitoring agents, etc) to the oVirt Node image
- 2 use cases
  - Change but still use with oVirt Engine
  - Use outside of oVirt Engine
- Need to provide rules for 'what can be changed' in order to prevent plugins from disrupting normal oVirt compatibility
- Some Challenges
  - Firewall rules
  - Service enablement
  - Initramfs regeneration
Roadmap – Monitoring Agents

• Enable agents like CIM providers, Matahari Agents, SNMP MIBs, etc
• Ideal candidates for plugin architecture
• This enables core image to be kept small, but users can add what they need for their specific mgmt infrastructure
• Matahari is likely more integral to oVirt, as guest agents and VDSM may eventually utilize QMF
Roadmap - Distribution Neutrality

• Currently heavily based on Fedora
• Is there desire for multiple oVirt Nodes based on various distributions?
• Challenges:
  • Need to move to generic configuration scripts
  • Node is livecd based which is not applicable to all distros, but there are functional equivalents
  • Need to abstract core oVirt Node functionality from distro specific, but still maintain a single ovirt-node code repo to prevent divergence
More information

- Mailing List: node-devel@ovirt.org
- IRC: #ovirt on OFTC
- Web Site: http://www.ovirt.org
- Git Repository: git://git.fedorahosted.org/git/ovirt/node.git
  - Moving soon to ovirt.org infrastructure
- Additional Info: http://fedorahosted.org/ovirt
  - Wiki contents in the process of being moved to http://ovirt.org/wiki
THANK YOU!

http://www.ovirt.org
Stateless - Configuration

- Bootstrapping
  - Embedding minimal configuration and certificates
    - initrd chaining
    - libguestfs to crack open appliance images
    - edit-livecd or similar tool
- Configuration Bundles
  - Contain files for overlay on rootfs or augeas/puppet scripts for processing
  - Optional: can be encrypted
  - Retrieval: keyed by some unique identified (MAC address)
    - Standard web/NFS server could be used
Stateless – Security

- Multiple solutions depending on concerns and/or regulations

- One solution:
  - Communication can be done over https using the server certificate to verify authenticity
  - Server can validate client by bundling a client certificate in the client image
  - Client certs can be datacenter specific or host specific
    - This choice directly correlates to the number of PXE images you need
Stateless – Bundle creation

- Offline tool to create configuration bundles for retrieval by clients, or...
- Capture configuration created by manual input or kernel command line to upload to config server
- Provide area for anonymous uploads from clients that are validated by administrator, encrypted and moved to config server download area