

RED HAT :: SAN DIEGO :: 2007

SUMMIT



Manageable Virtual Appliances

David Lutterkort

2007-05-09

Overview

- Virtual Appliances
 - Benefits
 - Problems
- Appliance Images
- Appliance Recipes
- Example



Appliances

- Virtual Appliance
 - one or more virtual machines packaged for distribution
 - supplier-defined function
- Distribution of a whole system
 - sealed black box (?)
- Typical distribution
 - One or more disks (binary images)
 - Example config file for virt platform
 - README or similar



Examples

- Routing firewall
- Asterisk VoIP
- Wiki
- Blogs
- Database server
- Tomcat
- LAMP stack



Appliances - Benefits

- Appliance installation simpler than application installation
- Appliance tested as a whole by supplier
- More latitude in OS and component selection
- Integrated, simplified configuration
- General virtualization benefits
 - Hardware isolation
 - Improved hardware utilization
 - Application isolation



Appliances - Problems

- Image size
- Safe deployment
- Scale problems
 - One appliance, two appliances, ...
 - Data center full of DSL routers ?



Appliances - Problems

- Separation
 - Keep user modifications separate from original appliance
- Configuration
 - Ease-of-use for consumers
 - Automation for enterprise users
- Discovery
 - Network config
 - Outgoing mail server



Builders and Users

- Development vs. Assembly
 - Collaboration application, developed as appliance
 - PostgreSQL + phpPgAdmin
 - Golden Master
- Consumers vs. Enterprise
 - Easy setup vs. automation
 - Small vs. large number of appliances
 - Single deployment vs. multiple deployments
 - No infrastructure vs. existing infrastructure



Virtualization Platforms

- libvirt
 - common API
 - abstract away platform differences
- Use libvirt capabilities
 - fully-virt vs. paravirt
 - Xen, kvm, qemu, ...
 - features such as pae vs. nonpae



Appliances vs. Bare-metal

- Similar problems and advantages as image-based management
 - Local modifications hard to capture
 - Updates clunky
- Stateless Linux
 - Run with almost entire filesystem readonly
 - Mark mutable files and categorize
 - Persistent (`/etc/statetab`)
 - Temporary (`/etc/rwtab`)



Images and Recipes

- Images: metadata + disk images
 - Easy to deploy
 - Updates unreliable
 - Hard to comprehend
- Recipes: metadata only
 - More deployment infrastructure
 - Complete description of contents
 - Easy to fit into existing infrastructure
- Hybrid approach
 - Build from recipe and create images



Images: VM descriptor

- Create libvirt XML from descriptor
- Check appliance matches platform
 - hypervisor (`xen`, `hvm`)
 - arch (`i386`, `x86_64`, ...)
 - features (`pae`, `nonpae`)
- Boot descriptor (multiple)
- Storage description
 - List disk images
 - Map into VM



Images: Disks

- Images are `raw` (would like to use `qcow`)
- Classify into `system/data/scratch`
 - Similar to classification for Stateless
- Replace system disks on update
- Multiple VM's can run same appliance
 - Must keep original appliance image and VM image separate
 - Track VM/appliance relation



Images: Building

- Create initial images in any way that seems convenient
 - For repeatability, use recipe
- Write/generate appliance descriptor
 - Multiple boot descriptors, e.g., paravirt and fully-virt
- Package as tarball
- Package as RPM
 - Use `yum` for distribution
 - Sign packages
 - Versioning
- Tools needed to simplify building



Images: Deploying

- Download and install into `/var/lib/appliances`
- Use `virt-install` to create VM
 - Check appliance matches host
 - Copy appliance disks and record VM <-> appliance association
 - Create libvirt XML and start VM
- Integration with `virt-manager` planned
 - Basis is `virt-install` API



Recipes

- Only metadata consisting of
 - Appliance descriptor
 - Kickstart for base system
 - Puppet manifest for appliance config
- Important parts of kickstart
 - Storage definition
 - Yum repos
- Keep kickstart simple, move appliance-specific setup to puppet



Recipes: Puppet

- *Manifest* describes configuration in declarative language
 - *Resources* describe basic config element
 - Files, users, services, packages, etc.
 - *Classes* group resources into logical units
 - webservers, mailserver, nagios-client
 - *Nodes* mapping classes to systems (hosts and guests)
- Client/server mode and command line mode
- Distribute puppet config as *module*



Recipes: Appliance Modifications

- *Overrides* of core appliance config
 - Closely related to appliance's function
 - Use site-specific SSL cert for appliance's webserver
 - Based on puppet's subclassing mechanism
- *Additions* to appliance config
 - Tangential to appliance's function
 - Send all `syslog` messages to central server
 - Map additional classes to VM's node
- Works around discovery problems



Recipes: Deploying

- Use `virt-factory`, integrates all necessary tools
- Simplest setup: `kickstart`, `virt-install`, `puppetmaster`
- Deployment driven by `virt-install`
 - Load puppet manifest into `puppetmaster`
 - Create and populate disk images with base system
 - Puppet client performs final setup/configuration



Example: Kronolith setup

- Simple web calendaring appliance based on Kronolith
 - Use PostgreSQL storage backend
 - Two disks
 - Main system disk
 - Data disk for PostgreSQL
- Create appliance by following Kronolith install guide
 - Create VM with minimal base system
 - Configure and initialize database
 - Open port 80 in firewall
 - ...



Example: Image and Recipe

- Produce Appliance Image
 - Write appliance descriptor
 - Compress disk images and package
 - Image ~ 320MB *compressed*
- Produce Appliance Recipe
 - Write appliance descriptor
 - Include kickstart file for base system
 - Write puppet module for kronolith setup
 - Use *cft* to automate this
 - Only text files – very small



Example: Mail setup

- Kronolith sends email reminders, needs mail server
- Solution for Appliance Image
 - User can configure mail server in Kronolith Web UI
 - Oops, modifies `/etc/horde`
 - Move `/etc/horde` to data disk
- Solution for Appliance Recipe
 - Make config file in `/etc/horde` template, include in recipe
 - User provides values for template
 - No original config information stored on appliance



Example: Disk Full

- How to indicate that data disk is filling up
 - Consumer/SMB happy with web UI or email notification
 - Useless for enterprise, wants monitoring
- How to expand data disk
 - Consumer/SMB needs help
 - Storage management in virt infrastructure



Links

ET Website with links to tools mentioned: <http://et.redhat.com/>

Mailing list: et-mgmt-tools@redhat.com

Commented image/recipe examples:

<http://people.redhat.com/dlutter/kronolith-appliance.html>

