Survey of Virtualization Tools in SUSE Linux Enterprise Server 11

Jim Fehlig
Software Engineer
jfehlig@suse.com

Mike Latimer
Software Engineer
mlatimer@suse.com
# Agenda

## Native hypervisor tools

### libvirt

### libvirt applications

- vm-install, virt-install, virt-clone
- vhostmd
- libvirt-cim
- virt-manager
- virt-viewer
- libguestfs
- virt-v2v
Native Hypervisor Tools
KVM

• qemu-kvm command line
  - qemu-kvm -m 1024 -smp 4 -name sles11sp3 ...

• QEMU monitor
  - Access with Ctr+Alt+2 (Ctr+Alt+1 to return to VM console)
  - 'help' for a list of all monitor commands, 'help <command>' for individual command usage

• qemu-img
  - Check, create, convert, snapshot, rebase, resize images
  - qcow2, vhd, vmdk, raw, qed, ...

• kvm_stat
  - Examine runtime statistics from kvm kernel module
Xen

• xm/xend
  - Default toolstack in SUSE Linux Enterprise Server 11 SP3
  - Legacy Xen toolstack
  - Aging, but stable and mature

• xl/libxl
  - New, light-weight toolstack developed by upstream Xen community
  - Technical preview in SUSE Linux Enterprise Server 11 SP3
  - Recommend disabling xend when using xl/libxl
  - Default toolstack in openSUSE13.1 and SUSE Linux Enterprise Server 12
Xen

- **xentop**
  - Display real-time information about the Xen host and domains

- **xenstore-**\{ls,read,rm,watch,write\}
  - CLI for xenstore database

- **Debug**
  - **xentrace / xentrace_format**
    - Capture Xen trace buffer data and format for human readability
  - **xenctx**
    - Inspect VM VCPU, register, and event state, optional stack trace
Linux Containers (LXC)

- LXC tools from linuxcontainers.org
  - Conf file to specify container settings
    - /usr/share/doc/packages/lxc/examples
    - lxc.conf man page
  - Template script used to configure the container
    - /usr/share/lxc/templates
  - lxc-create -t TEMPLATE -f CONF -n NAME
  - lxc-{start, console, stop, destroy} -n NAME
  - man lxc, man lxc-subcommand
Linux Containers (LXC)

- yast2-lxc
  - Yast module that wraps the lxc tools
- libvirt-lxc
  - Not fully supported in SUSE Linux Enterprise Server 11 SP3
  - Setup performed in TEMPLATE done by libvirt
  - libvirt domXML instead of lxc.conf
  - Preferred userspace going forward (SUSE Linux Enterprise Server 12 family)
libvirt
libvirt

• Stable API or managing virtualization on a host
  – Storage, network interfaces, networks, host devices, hypervisors, and virtual machines

• XML schema for describing configuration of managed entities

• Wide hypervisor support
  – KVM/QEMU, Xen, LXC available in SUSE Linux Enterprise Server
  – KVM/QEMU, Xen, LXC, ESX, Hyper-V, XenServer, VirtualBox, and UML available in openSUSE

• Used by a wide variety of tools and products
  – http://libvirt.org/apps.html
libvirt Architecture

stateful

Client (virsh)

remote

libvirt.so

ESX

hyperv

RPC

stateless

libvirtd

qemu

xen

lxc

storage

network

VMware

Hyper-V
libvirt

• Benefits
  - Normalized API for managing virtual machines
  - Stable API and configuration format (XML)
  - Insulate users from changes in underlying components
  - Secure migration protocols
  - Integration with other subsystems used in the virtualization ecosystem
libvirt

- Integration with cgroups
  - Isolation and resource control (tuning)
- Integration with AppArmor, SELinux, DAC
  - Resource confinement
- Integration with lock managers such as virtlockd and sanlock
  - Resource protection
- Audit logging
  - Resource inspection
libvirt

- Storage
  - Pools and volumes

- Networks
  - Virtual networks

- Network Interfaces
  - Physical devices, bridges, bond, VLANs
libvirt + cgroups

- cgroups provides grouping of processes and a means of applying controls to those groups
- cgroup controllers provide the controls via a virtual file system interface
  - Isolation controllers: cpuset, freezer, devices
  - Resource controllers: cpu, cpuacct, memory, blkio
- Init system mounts control groups file systems
  - /etc/init.d/cgconfig start
    - /cgroups/$CONTROLLER-NAME
  - /etc/gcconfig.conf to adjust which controllers are mounted and their mount point
libvirt + cgroups

• cgroup controllers
  - cpuset: CPU and NUMA node affinity
  - cpu: scheduler tunables
  - memory: memory tunables
  - cpuacct: CPU statistics
  - devices: device whitelisting
  - blkio: block device IO tunables

• QEMU driver
  - Will use all mounted controllers by default
  - Explicit control via 'cgroup_controllers' setting in /etc/libvirt/qemu.conf
libvirt + cgroups

• LXC driver
  - cpuset, devices, memory controllers are mandatory
  - cpu, cpuatct, blkio controllers are optional and used if mounted

• VMs placed in their own group by default
  - /cgroups/<controller>/machine/<vm-name>.libvirt-qemu

• Custom grouping is supported
  <resource>
    <partition>/machine/production</partition>
  </resource>
  /cgroups/<controller>/machine/production.partition
libvirt + cgroups + cpuset controller

• Bind processes in a group to a set of CPUs
  - Recommended in NUMA environments to prevent a VM's vCPUs from crossing NUMA nodes

• Query topology
  - virsh nodeinfo
  - virsh capabilities
    - NUMA info described in <topology> element
  - virsh freecell <numa_node>

• XML configuration
  - <vcpus cpuset='0-3'>4</vcpus>

• Runtime tuning
  - virsh vcpupin sles11sp3 0 4
libvirt + cgroups + cpu controller

- Control priority (nice level) of processes in a group
- Settings exposed via virsh schedinfo
- Query
  - virsh schedinfo sles11sp3
- XML configuration
  <cputune>
    <shares>1024</shares>
  </cputune>
- Runtime tuning
  - virsh schedinfo sles11sp3 --set cpu_shares=1 --set vcpu_period=1000 --set vcpu_quota=10000 --set emulator_period=1000 --set emulator_quota=10000
libvirt + cgroups + memory controller

- Limit RAM and swap usage of processes in a group
- Settings exposed via virsh memtune command
- Query
  - `virsh memtune sles11sp3`
- XML configuration
  ```xml
  <memtune>
    <hard_limit unit='G'>2</hard_limit>
  </memtune>
  ```
- Runtime tuning
  - `virsh memtune sles11sp3 --hard-limit 4G --soft-limit 2M
    --swap-hard-limit 6G`
libvirt + AppArmor

• Basic protection of host from malicious VMs
  ‒ Default if AppArmor is enabled
  ‒ libvirtd is confined based on /etc/apparmor.d/usr.sbin/libvirtd

• sVirt protection extends basic to include inter-VM confinement
  ‒ /etc/libvirt/qemu.conf
    ‒ security_driver = “apparmor”
  ‒ AppArmor profile granting access to resources is generated when starting VM
  ‒ Custom profiles can be placed in /etc/apparmor.d/libvirt/libvirt-<vm_uuid>
libvirt + SELinux

• SELinux is supported in SUSE Linux Enterprise Server 11 SP3, but no profiles are provided

• Basic protection of host from malicious VMs
  - Default if SELinux is enabled on the host
  - Currently, burden on the user to create SELinux profile
  - All QEMU instances placed in root:system_r:qemu_t domain

• sVirt protection extends basic to include inter-VM confinement
  - /etc/libvirt/qemu.conf
    - security_driver = “selinux”
  - QEMU instances placed in their own domain
libvirt + DAC

• Default security driver in QEMU driver
  – Built into the driver, cannot be disabled

• Uses Linux's Discretionary Access Control
  – Run QEMU processes as specified user:group
  – /etc/libvirt/qemu.conf
    – user = “qemu”
    – group = “qemu”

  – SUSE Linux Enterprise Server 11 default is user 'root' and group 'root' for compatibility between service packs

  – SUSE Linux Enterprise Server 12 default will by user 'qemu', group 'qemu'
**libvirt + virtlockd**

- Virtlockd lock manager is included in libvirt and promoted by libvirt community
- Provides resource protection via fcntl()-based locks
- Requires shared file system
- Locks created automatically by libvirt
- `/etc/libvirt/qemu.conf`
  - `lock_manager = “lockd”`
- Customization via `/etc/libvirt/qemu-lockd.conf`
- virtlockd can be updated without terminating VMs
libvirt + virtlockd Architecture

Client

remote ESX libvirt.so hyperv test

RPC

libvirtd

lxv xen qemu storage network

RPC

QEMU virtlockd

shared file system
libvirt + sanlock

- Locks are actually leases in sanlock
- Provides resource protection via disk paxos algorithm
- Preferred use with SAN
- Use of shared file system possible, but discouraged
- Locks (leases) created automatically by libvirt
- VM is “fenced” on lease failure
  - sigterm of QEMU process, followed by sigkill if needed
- /etc/libvirt/qemu.conf
  - lock_manager = “sanlock”
- Terminate VMs before updating sanlock
libvirt + Audit

- Integration with Linux audit subsystem
- Audit notification when VM consumes host resources
  - Start, stop, save, restore, etc
  - Device hotplug
  - Memory balloon
  - vCPU modification
- /var/log/audit/audit.log
libvirt + Storage

- Storage pools and volumes
  - Volumes created from pools, providing block devices for VMs
  - Volume can be a block device or image file

- Storage pools
  - Directory
    - Manage images in a directory
  - File system
    - Mount file system and manage images in the mount point
  - Network
    - Mount remote file system and manage images in the mount point
libvirt + storage

• Storage Pools
  – LVM
    - Manage images/block devices in an LVM volume group
  – Disk
    - Manage volumes on a physical disk (partition = volume)
  – iSCSI
    - Manage volumes on an iSCSI target
  – SCSI
    - Manage pre-existing SCSI LUNs on a SCSI HBA
libvirt + storage

• Storage pool configuration
  
  <pool type='dir'>
  <name>vm-images</name>
  <target>
    <path>/var/lib/libvirt/images</path>
  </target>
  </pool>

• Storage pool management
  
  – virsh pool-define <pool.xml>; virs pool-start <pool-name>
  – virsh help pool
libvirt + storage

• Storage volume configuration
  
  ```xml
  <volume type='file'>
    <name>sles11sp3</name>
    <capacity unit='G'>20</capacity>
    <target>
      <path>/var/lib/libvirt/images/sles11sp3.qcow2</path>
      <format type='qcow2'/>
    </target>
  </volume>
  ...
  </volume>
  ```

• Storage volume management
  
  - `virsh vol-create <vol.xml>`
  - `virsh help volume`
libvirt + networking

• Network configuration

  <network>
  <name>default</name>
  <bridge name='virbr0'/>
  <forward />
  <ip address='192.168.122.1 netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2 end='192.168.122.254'/>
    </dhcp>
  </ip>
  </network>

• Network management

  - virsh network-define <network.xml>; virsh network-start <name>
  - virsh help network
libvirt + Host Network Interfaces

• libnetcontrol used to manage host network interfaces
  - Ethernet devices
  - Bridges
  - Bonds
  - VLANs
libvirt + Host Network Interfaces

• Network Interface Configuration

```xml
<interface type='bridge' name='br0'>
  <start mode='onboot'/>
...
<bridge stp='off'>
  <interface type='ethernet' name='eth0'>
    <mac address='aa:bb:cc:dd:ee:ff'/>
  </interface>
</bridge>
</interface>
```

• Network Interface Management
  - `virsh iface-define <iface.xml>; virsh iface-start <iface-name>`
  - `virsh help interface`
libvirt Applications
vm-install

- Virtual machine installation tool created and maintained by SUSE
- Default installation tool used by virt-manager
- VM installation on local host only
- Command Line Interface (CLI) and graphical mode

- CLI
  - vm-install --name=sles11sp3 --vcpus=4 --memory=1024 --disk=/var/lib/libvirt/images/sles11sp2.qcow2,vda,qcow2 --nic=bridge=br0,mac=aa:bb:cc:dd:ee:ff ...
  - Don't forget about --debug, --preserve-on-error, and --no-install options

- man vm-install
**virt-install**

- Virtual machine installation tool created and maintained as part of community virt-tools project
- Optional installation tool for virt-manager
- VM installation on local or remote host
- Command Line Interface (CLI) providing an easy way to provision operating systems into virtual machines
  
  ```
  virt-install --virt-type kvm --name sles11sp3 --ram 1024 --disk path=/var/lib/libvirt/images/sles11sp3.img,size=8  --graphics vnc --cdrom /dev/cdrom
  ```

- man virt-install
virt-clone

- Virtual machine clone tool created and maintained as part of community virt-tools project
- Copies VM disk images and defines a new VM with identical hardware configuration
- Command Line Interface for rapid cloning of virtual machine images
- `virt-clone --original=origVMName
 --name=newVMName --mac=aa:bb:cc:dd:ee`
- man virt-clone
virt-manager

- Desktop application for managing the host and virtual machines through libvirt
- Summary view of running virtual machines
- Performance and resource utilization statistics
- Wizards to create networks, storage pools, storage volumes, and network interfaces
- Wizards to enable creating new virtual machines and modifying existing ones
virt-viewer

- Lightweight interface for interacting with graphical display of virtual machines
- Supports Virtual Network Computing (VNC) and Simple Protocol for Independent Computing Environments (SPICE)
- `virt-viewer [--connect=URI] vm-name`
vhostmd

• Provides a “metrics communication channel” between the virtual machine host and its hosted virtual machines

• Allows limited introspection of host resource usage from within virtual machines

• Configured via /etc/vhostmd/vhostmd.conf

• Metrics written to a block device that is provided read-only to VMs

• Use Case: Inspect host state when faced with VM QoS issues
libvirt-cim

- libvirt-based implementation of the Distributed Management Task Force (DMTF) Common Information Mode (CIM) virtualization standards
- Essentially a CIM binding for libvirt
- Traditionally used by Tivoli, CA Unicenter, and similar enterprise management systems
libguestfs

• Set of tools for accessing and modifying virtual machine disk images

• Supports all types of Linux file systems
  – Ext2/3/4, XFS, btrfs, etc

• Supports Windows file system
  – VFAT and NTFS

• Supports Mac OS X and BSD file systems

• Supports many disk image formats
  – Raw, qcow2, VMDK, VHD/VHDX
libguestfs

• Includes several useful tools
  - guestfish, guestmount, virt-cat, virt-copy-in, virt-copy-out, virt-df, virt-edit, virt-format, virt-inspector, virt-resize, virt-sparsify, etc

• Provides a library for use in your custom applications and includes several language bindings
  - Perl, Python, Ruby, Java, etc.
  - Note – guestfs tools work differently depending on who you are logged in as!
    - root: qemu:///system
    - user: qemu:///session
guestfish

• Safe shell for examining and manipulating virtual disk images
  - Exposes all capabilities of guestfs api
    - Uses kernel and initrd found in /usr/lib64/guestfs

• Usage:
  - # guestfish
    >&<fs> add_drive "/var/lib/libvirt/images/TestPool/disk0.qcow2"
    >&<fs> launch
    >&<fs> inspect-os
    /dev/sda2
    >&<fs> inspect-get-product-name
    SUSE Linux Enterprise Server 11 (x86_64)
    >&<fs> mount /dev/sda2 /
    >&<fs> sh “cat /etc/fstab | grep sda2”
    /dev/sda2 / ext3 acl,user_xattr 1 1
virt-rescue

• Rescue shell for virtual disk images
  - Provides an interactive shell which supports standard Linux commands
    - Uses kernel and initrd found in /usr/lib64/guestfs
  
• Usage:
  - # virt-rescue -a /var/lib/libvirt/images/TestPool/disk0.qcow2
    > <rescue> mount /dev/sda2 /sysroot
    > <rescue> cat /sysroot/etc/fstab
    /dev/sda2  /  ext3  acl,user_xattr  1  1
Managing Disk Space with guestfs

- **virt-df**
  - Shows disk usage for each partition on a virtual disk

- **virt-filesystems**
  - Shows file systems, partitions, block devices, and LVM info

- **virt-resize**
  - Resize (shrink or expand) a file system
    - Create a new disk using 'truncate' first, then virt-resize into the new disk

- **virt-sparsify**
  - Converts a disk image to a sparse file
virt-v2v

• Converts virtual machines running on a “foreign” hypervisor to run on KVM
  - Source hypervisors:
    - Xen, VMware, VirtualBox
  - Supported guests:
    - SLES, openSUSE
    - RHEL, Fedora (and various clones)
    - Windows

• Usable with SUSE Linux Enterprise Server 11 SP3, but not fully supported until SUSE Linux Enterprise Server 12
**virt-v2v** (Continued)

- **Process overview:**
  - Virtual disks are copied to target KVM host
    - Original disk images are not modified
  - Image is inspected and destination capabilities are determined
  - Any required changes are made to the new image
    - virtio kernel is installed, serial console changed, etc
  - New guest is created

- **Sample command line**
  - virt-v2v -ic xen+ssh://root@192.168.1.120 -os XenStore SLES11-SP3
Related Sessions

**TT1455** – Using Linux Containers as a Virtualization Option

**TT1469** – Securing your Xen Virtualization Environment

**TT1474** – SUSE Linux Enterprise Server as a Virtualization Host
Have fun with the virtualization stacks in SUSE Linux Enterprise Server

www.suse.com/solutions/platform.html#virtualization

Thank you.
Unpublished Work of SUSE. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.