



Learn about KVM virtualization with QEMU and Libvirt

NetApp virtualization solutions

NetApp
November 07, 2025

Table of Contents

- Learn about KVM virtualization with QEMU and Libvirt 1
 - Overview of components 1
 - Cluster Management 2
 - Compute 2
 - Storage 4
 - Network 5
 - Monitoring 6
 - Data Protection 6

Learn about KVM virtualization with QEMU and Libvirt

Learn about KVM virtualization and how it works with QEMU and Libvirt to create a comprehensive Linux virtualization platform. Discover how these components enable near-native performance virtual machines while providing flexible management capabilities for compute, storage, and networking resources in enterprise environments.

Overview of components

1. KVM (Kernel-based Virtual Machine):

- Role: KVM is a kernel module that provides the core virtualization capabilities by leveraging hardware-assisted virtualization extensions in the CPU (e.g., Intel VT-x or AMD-V).
- Functionality: KVM allows the Linux kernel to act as a type-1 hypervisor, enabling it to create and manage virtual machines with near-native performance for CPU and memory operations.
- Lowest Layer: KVM is the lowest level component, interacting directly with the hardware to provide the necessary infrastructure for full virtualization.

2. QEMU (Quick EMUlator):

- Role: QEMU is a machine (hardware) emulator that provides the other virtualized hardware components for the guest system, such as network interfaces, disk controllers, and graphics cards.
- Functionality: QEMU can operate as a type-2 hypervisor when KVM is not available, but it's significantly slower as it has to simulate the guest CPU in software. When combined with KVM, QEMU uses KVM to accelerate CPU and memory operations, while QEMU handles the emulation of peripherals and other hardware devices.
- Interaction with KVM: QEMU interacts with KVM through a device file (e.g., `/dev/kvm`) using `ioctl()` system calls to manage virtual machine processes and communicate with KVM.

3. [Libvirt](#):

- Role: Libvirt is a virtualization library and API that provides a higher-level management interface for various virtualization platforms, including KVM/QEMU, Xen, and VMware ESXi.
- Functionality: Libvirt simplifies the management of virtual machines by providing a unified interface and set of tools. It acts as an abstraction layer, allowing users and applications to manage VMs without needing to interact directly with the underlying hypervisor's specific commands or APIs.
- Key Features:
 - VM lifecycle management: Start, stop, pause, save, restore, and migrate VMs.
 - Remote management: Control VMs on remote hosts over SSH or other protocols.
 - Storage management: Create and manage storage pools and volumes for VMs.
 - Virtual networking: Configure virtual networks with NAT, bridging, and other modes.
 - Security: Integrate with SELinux and AppArmor for security confinement of VMs.
 - Hotplug: Add or remove devices like disk and network interfaces while the VM is running.
 - Tools: Libvirt includes command-line tools like `virsh` and graphical tools like `virt-manager` for managing VMs.
- How they work together:

- KVM: Provides the kernel-level virtualization infrastructure.
 - QEMU: Provides the emulated hardware and manages the VM processes.
 - Libvirt: Acts as the management layer, providing an API and tools to control KVM/QEMU and other hypervisors.
- In essence: KVM provides the hardware acceleration for virtualization, QEMU provides the emulated hardware and runs the VM, and Libvirt provides the user-friendly management interface and API to control the entire setup.

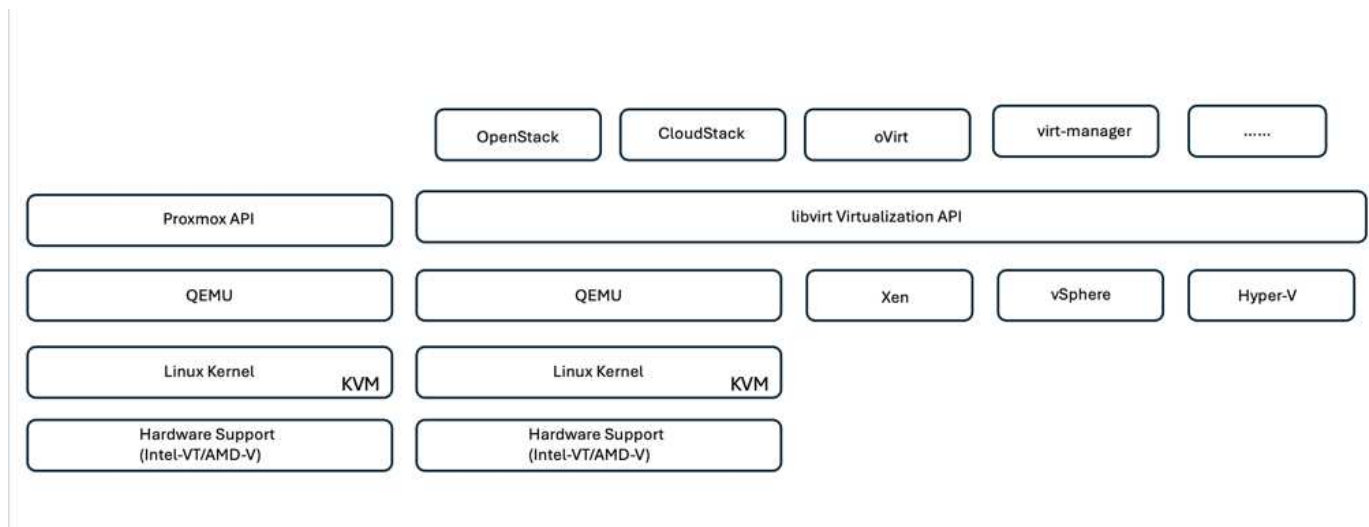
Libvirt client tools can be used to manage VMs or operations from CLI, GUI or web based using cockpit machines. For list of applications using libvirt, check [here](#).

As applications move into Kubernetes environment, take a look into Kubevirt to run virtual machines as pods in those environments.

[Ansible module](#) is available for [libvirt](#) for automation purposes.

Cluster Management

Typically Virtualization Manager or virsh cli tool manages single host at a time. To manage multiple hosts in a cluster, higher level applications like oVirt, CloudStack or OpenStack is often used. Those tools will assist in placement of VMs and distribute the load. If looking for high availability of certain VMs in a small sized cluster environment, Pacemaker along with Corosync is used or check your management stack option.



Compute

Libvirt provides a comprehensive framework for managing virtual machine resources and functionality. This includes tasks related to:

1. Virtual Machine (Domain) Management:

- Lifecycle Operations: Libvirt offers a complete set of operations for managing the state of virtual machines (referred to as "domains" in libvirt terminology). This includes starting, stopping, pausing, resuming, saving, restoring, and migrating VMs.
- XML Configuration: Virtual machine configurations are defined using XML files. You can use tools like virsh or virt-manager to create, modify, and delete these XML configurations.

- Remote Management: You can manage virtual machines on remote hosts using libvirt's remote protocol, which supports various network transports like SSH.

2. Resource Allocation and Management:

- CPU Management: Libvirt allows you to configure guest CPUs, including specifying the number of virtual CPUs, controlling CPU pinning (associating vCPUs with specific physical CPUs on the host), and managing CPU modes (like host-passthrough to expose the host's CPU features to the guest).
- Memory Management: You can allocate memory to virtual machines and configure memory overcommit (allowing the sum of assigned memory to VMs to exceed the physical memory available on the host).
- Storage Management: Libvirt can manage various types of storage for virtual machines, including disk images (in formats like qcow2, vmdk, and raw), NFS shares, LVM volume groups, iSCSI shares, and raw disk devices.
- Host Device Management: You can manage physical and virtual host devices like USB, PCI, SCSI, and network devices, including virtualization capabilities like SR-IOV and NPIV.

3. Virtual Networking:

- Virtual Network Switches: Libvirt creates virtual network switches (bridges) to connect virtual machines to each other and to the host network.
- Networking Modes: It supports various networking modes, such as NAT, bridged, isolated, and routed, to configure how VMs interact with the network.
- Firewall Rules: Libvirt automatically manages firewall rules (using iptables) to control network traffic for virtual networks.

4. Performance Optimization:

- CPU Pinning: Pinning vCPUs to specific physical CPUs can improve cache efficiency and performance, especially in NUMA environments.
- NUMA Tuning: You can optimize performance on NUMA systems by limiting guest size to the amount of resources on a single NUMA node and pinning vCPUs and memory to the same physical socket that's connected to the I/O adapter.
- Hugepages: Using hugepages can improve performance by reducing the overhead associated with managing small memory pages.

5. Integration with Other Tools:

- virsh: The command-line interface for interacting with libvirt.
- virt-manager: A graphical tool for managing virtual machines and libvirt resources.
- OpenStack: Libvirt is a commonly used virtualization driver in OpenStack.
- Third-party tools: Many other tools and applications leverage libvirt's API for managing virtual machines, including cloud management platforms and backup solutions.

KVM hypervisor allows to overcommit CPU and memory as typically VM guests are under utilized. But needs to be monitored and balance it out for better performance.

The VM metadata is stored as XML in `/etc/libvirt/qemu`. The VM can be created using `virt-install` or `virsh cli`. Virt-Manager can be utilized if UI is preferred or use the upper management stack.

In summary, libvirt provides a comprehensive management layer for the compute aspects of virtualization, allowing you to control virtual machine lifecycles, allocate resources, configure networking, optimize performance, and integrate with other tools and platforms.

Storage

The VM disks can be dynamically provisioned on storage pool or can be pre-provisioned for the VM by the storage administrator. There are various pool types supported by libvirt. Here is the list of applicable pool types along with supported storage protocol.

The popular choice is dir. Then netfs and logical. iscsi and iscsi-direct uses single target and doesn't provide fault tolerance. mpath provides multipath but not dynamically allocated. It is used more like raw device mapping in vSphere. For file protocols (NFS/SMB/CIFS), mount options can be specified in [automounter](#) or fstab and dir pool type is used. In case of block protocols (iSCSI,FC,NVMe-oF), shared filesystem like ocfs2 or gfs2 is used.

Storage Protocol	dir	fs	netfs	logical	disk	iscsi	iscsi-direct
mpath	SMB/CIFS	Yes	No	Yes	No	No	No
No	No	NFS	Yes	No	Yes	No	No
No	No	No	iSCSI	Yes	Yes	No	Yes
Yes	Yes	Yes	Yes	FC	Yes	Yes	No
Yes	Yes	No	No	Yes	NVMe-oF	Yes	Yes

Notes:

1 - Additional configuration might be required.

Based on storage protocol used, additional packages needs to be available on the host. Here is the sample list.

Storage Protocol	Fedora	Debian	pacman
SMB/CIFS	samba-client/cifs-utils	smbclient/cifs-utils	smbclient/cifs-utils
NFS	nfs-utils	nfs-common	nfs-utils
iSCSI	iscsi-initiator-utils,device-mapper-multipath,ocfs2-tools/gfs2-utils	open-iscsi,multipath-tools,ocfs2-tools/gfs2-utils	open-iscsi,multipath-tools,ocfs2-tools/gfs2-utils
FC	sysfsutils,device-mapper-multipath,ocfs2-tools/gfs2-utils	sysfsutils,multipath-tools,ocfs2-tools/gfs2-utils	sysfsutils,multipath-tools,ocfs2-tools/gfs2-utils
NVMe-oF	nvme-cli,ocfs2-tools/gfs2-utils	nvme-cli,ocfs2-tools/gfs2-utils	nvme-cli,ocfs2-tools/gfs2-utils

Storage pool details are stored in XML file at /etc/libvirt/storage.

To import VM data from a vSphere environment, look into [Shift toolkit](#).

Network

Libvirt provides robust virtual networking capabilities for managing virtual machines and containers. It achieves this through the concept of a virtual network switch or bridge.

Core Concepts:

* Virtual Network Switch (Bridge): This acts like a software-based network switch on your host server. Virtual machines connect to this switch, and traffic flows through it.

* TAP Devices: These are special network devices that function as the "virtual cables" connecting the virtual machine's network interface to the libvirt bridge.

- Networking Modes: Libvirt supports various network configurations to meet different needs:
 - NAT (Network Address Translation): This is the default mode. VMs connected to a NAT network can access the external network using the host's IP address, but external hosts cannot directly initiate connections to the VMs.
 - Bridged: In this mode, the virtual network is directly connected to the same network segment as the host. This allows VMs to appear as if they are directly connected to the physical network.
 - Isolated: VMs on an isolated network can communicate with each other and the host, but they cannot reach anything outside the host. This is useful for testing or secure environments.
 - Routed: Traffic from the virtual network is routed to the physical network without NAT. This requires proper routing configuration on the host's network.
 - Open: Similar to Routed mode, but without any firewall rules automatically applied by libvirt. This assumes that network traffic will be managed by other systems.
- DHCP and DNS: Libvirt can manage DHCP services for its virtual networks using dnsmasq, allowing it to assign IP addresses to VMs and handle DNS resolution within the virtual network.
- Firewall Rules: Libvirt automatically sets up iptables rules to control traffic flow for virtual networks, particularly in NAT mode.

Managing Libvirt Networks:

- virsh: The virsh command-line tool provides a comprehensive set of commands for managing virtual networks, including listing, starting, stopping, defining, and undefining networks.
- Virtual Machine Manager (virt-manager): This graphical tool simplifies the creation and management of virtual networks with an intuitive user interface.
- XML Configuration: Libvirt uses XML files to define the configuration of virtual networks. You can edit these XML files directly or use tools like virsh net-edit to modify network configurations.

Common Use Cases:

- NAT: Simple, basic connectivity for VMs on a host with a single network interface.
- Bridged: Integrating VMs seamlessly into an existing network.
- Isolated: Creating secure or testing environments where VMs are restricted from external access.
- Routed: More advanced scenarios where specific routing is required.
- Open vSwitch (OVS): For complex, large-scale deployments requiring advanced network management and automation.

By leveraging these features, libvirt provides a flexible and powerful framework for managing virtual machine networking in Linux environments.

Monitoring

NetApp Data Infrastructure Insights (formerly Cloud Insights) is a cloud-based infrastructure monitoring and analytics platform that provides comprehensive visibility into your IT infrastructure, including virtual machines.

While Data Infrastructure Insights is known for its strong focus on monitoring NetApp storage and VMware environments, it also has capabilities for monitoring other types of infrastructure and workloads.

Here's how you can potentially monitor Libvirt-based virtual machines with NetApp Data Infrastructure Insights:

1. Data Collectors:

- Data Infrastructure Insights operates through Acquisition Unit software, which uses various data collectors to gather data from your infrastructure.
- Data Infrastructure Insights has collectors for heterogeneous infrastructure and workloads, including Kubernetes.
There's also an open Telegraf collector and open APIs for easy integration with other systems.

2. Potential Integration with Libvirt:

- Custom Data Collection: You could potentially use the open Telegraf collector or the Data Infrastructure Insights API to collect data from your Libvirt-based systems. You would need to write or configure the collector to gather metrics from Libvirt using its API (e.g., via the virsh commands or by accessing Libvirt's internal metrics).

3. Benefits of Monitoring Libvirt with Data Infrastructure Insights:

- Unified Visibility: Gain a single view of your virtualized environment, including both your NetApp storage and your Libvirt-based VMs.
- Performance Monitoring: Identify performance bottlenecks and resource constraints, whether they're internal to the VMs or related to the underlying infrastructure supporting them.
- Resource Optimization: Analyze workload profiles to right-size VMs, reclaim unused resources, and optimize resource utilization across your environment.
- Troubleshooting: Quickly identify and resolve issues by correlating VM performance metrics with back-end storage metrics for end-to-end visibility.
- Predictive Analytics: Use machine learning for intelligent insights and to proactively identify potential issues before they impact performance.

In summary, while Data Infrastructure Insights has strong support for VMware, it's possible to integrate it with Libvirt-based virtualization by using custom data collectors or leveraging its open APIs. This would provide unified visibility, enhanced performance monitoring, and resource optimization capabilities for your Libvirt environment within the Data Infrastructure Insights platform.

Data Protection

Protecting data for Libvirt-based virtual machines with NetApp ONTAP can be achieved through several methods, often leveraging ONTAP's built-in data protection features. Here's a breakdown of common strategies:

1. Using ONTAP's Native Data Protection Features:

- Snapshots: ONTAP's core data protection technology is Snapshots. These are fast, point-in-time copies of your data volumes that require minimal disk space and have negligible performance overhead. You can use Snapshots to create frequent backups of your Libvirt VM disks (assuming they are stored on ONTAP volumes).

- SnapMirror: SnapMirror is used to asynchronously replicate Snapshot copies from one ONTAP storage system to another. This allows you to create disaster recovery (DR) copies of your Libvirt VMs at a remote site or in the cloud.
- SnapVault: SnapVault is used to back up data from multiple storage systems to a central ONTAP system. This is a good option for consolidating backups of many Libvirt VMs from different hosts onto a central backup repository.
- SnapRestore: SnapRestore allows you to quickly restore data from Snapshot copies. This is essential for recovering your Libvirt VMs in the event of data loss or corruption.
- FlexClone: FlexClone creates writable copies of volumes based on Snapshot copies. This is useful for quickly creating test/development environments based on production VM data.
- MetroCluster/SnapMirror active sync: For automated zero-RPO (Recovery Point Objective) and site-to-site availability, you can use ONTAP MetroCluster or SMas, which enables to have stretch cluster between sites.

2. Integration with Third-Party Backup Solutions:

Many third-party backup solutions integrate with NetApp ONTAP and support backing up virtual machines. You can use these solutions to back up your Libvirt VMs to ONTAP storage, leveraging ONTAP's data protection features. For example, some backup solutions use ONTAP's Snapshot technology for fast, agentless backups.

3. Scripting and Automation:

You can create scripts to automate the process of creating ONTAP Snapshots of your Libvirt VM volumes. These scripts can leverage ONTAP's command-line interface or APIs to interact with the storage system.

Key Considerations:

- Storage Location: Your Libvirt VM disk images should be stored on ONTAP volumes to leverage ONTAP's data protection features.
 - Network Connectivity: Ensure network connectivity between your Libvirt hosts and your ONTAP storage system.
 - HBA Management: If using Fibre Channel (FC) for storage connectivity, ensure you have the necessary HBA management packages installed on your Libvirt hosts.
 - Monitoring and Reporting: Monitor your data protection operations and ensure they are completing successfully.
- By combining Libvirt's capabilities with ONTAP's robust data protection features, you can implement a comprehensive data protection strategy for your virtualized environment.

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.